

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

APPENDIX III

(NASA-CF-144558) ENGINEERING STUDIES OF
VECTOCARDIOGRAPHS IN BLOOD PRESSURE
MEASURING SYSTEMS, APPENDIX 3 Final Report
(Trustees of Health and Hospitals of the
City) 210 p HC \$7.75

N76-12698

Unclas
01916

CSCI 06E G3/52



APPENDIX III

- A. ASM80 Manual by Joseph B. Walters, Jr.
- B. Signal Preprocessing as an Aid to On-Line EKG Analysis
by Joseph B. Walters, Jr.
- C. High Speed Evaluation of Magnetic Tape Recordings of
Electrocardiograms by Peter B. Kurnik

ASM80 MANUAL

By

Joe Walters, Jr.

Copyright, © 1974

By

S. K. Burns

All Rights Reserved

ORIGINAL PAGE IS
OF POOR QUALITY

1. INTRODUCTION

ASM80 is a two pass symbolic assembly program for the INTEL 8080 Microprocessor. The assembler runs on the NOVA line of mini-computers under DOS or RDOS.

ASM80 accepts symbolic assembly language statements as input, producing absolute binary as output. The output is suitable for input to the FROM 1702A programmer, and the standard binary loader for the 8080 system.

In describing statements in this manual, the following conventions will be used:

- 1) All numbers are decimal unless explicitly marked otherwise. A number followed by the character X or H, indicates that the number is expressed in the hexadecimal radix, as 16X is 16 (decimal). Octal numbers are indicated by a terminating letter K as the final character as 16K, which is 8 decimal.
- 2) In statement descriptions, items enclosed in square brackets are optional.
- 3) Items enclosed in wavy brackets "{}" are optional and may be repeated any number of times.
- 4) The symbol CR indicates a carriage return character.
- 5) In those cases where spacing characters are mandatory in a statement, the symbol "Δ" will be used to indicate any number of blanks or tabs in the line. Unless so noted, spaces are not significant.

1.1 ASM80 Language

ASM80 programs are generally prepared on a NOVA system, using the standard editor. Any other source of ASCII input, which can be loaded on a NOVA system as a text file, and which conforms to the programming conventions outlined in the section is acceptable as input to the Assembler.

Each statement is written on a single line, terminated by a carriage return. Blank lines are allowed so the user may group statements. Statements are format free; that is, individual elements of a statement need not appear in parti-

cular columns of the input line. The elements of statements are defined by specific delimiter characters.

There are four types of elements in an ASM80 statement. The assembler identifies the element type by its relative location, and by the presence of special delimiters. The individual elements need not appear in specific columns, but their order is fixed. Statements have the general form:

label: operator operand1,operand2 ;comments(carriage return)

Each statement is proceeded by the assembler as required for the operator given by the programmer. Some operators result in the generation of binary machine code, while others are simply interpreted by the assembler to control the assembly process.

The second type of ASM80 statement does not have an operator, but rather consists of a list of expressions. In general, the format of such an expression statement is:

label: expression1,expression2,.....expressionN ;comment(carriage return)

Expressions will be discussed in detail later, but they may be as simple as literal numbers. An example of this would be:

BITMSK: 01X,02X,04X,08X,10X,20X,40X,80X

This statement generates 8 bytes of data, corresponding in sequence to the 8 simple expressions in the above statement. The symbol BITMSK is assigned the value of the byte address of the first byte in this table.

By default, the assembler generates a single byte of output for each expression in the list. The psuedo-op "EXPIEN" allows the programmer to change the default output length to

ORIGINAL PAGE IS
OF POOR QUALITY

1,2,3, or 4 bytes. See Section for details.

The third type of statement allows the user to assign a value to a symbol without using the symbol as a label. This statement is the assignment statement, and the general form is:

symbol1,symbol2,...symbolN = expression ;comment(carriage return)

The assembler simply evaluates the single expression given, then assigns the value of the expression to the symbols given in the list to the left of the equals sign. An example of a possible expression is:

TOP.USED.CORE = TOP.USED.CORE + 8

This statement reassigns a new value to the symbol "TOP.USED.CORE" which is its old value plus 8.

The symbols which are assigned a value may not be used as labels in the program since this would provide conflicting information about the location of the statement specified by the label.

1.1.1 Labels

A label is a symbol, created by the programmer to identify a particular statement in his program. A label must be a properly formed symbol (see section 1.2) followed by a colon, ":". While our examples of a statement have only showed one label, it is legal to define multiple labels for a single statement, i.e.:

label1: label2: label3: operator operand (Carriage Return)

In practice, the assembler assigns a value to a label when it is defined. This value is the byte address, in INTEL memory, of the first byte of the binary generated by the statement following the label. Thus, every label

is assigned a value between U and FFFFH inclusive.

1.1.2 Operator

An operator may be one of the INTEL 8080 machine instructions defined in the specifications for the 8080 microprocessor, or a command for the assembler. A command for the assembler is often referred to as a psuedo-operation, or psuedo-op.

The operator determines the format of the remainder of the statement, such as the number and type of operands.

1.1.3 Operands

Operands are items which specify the data item, or the address of the data item, to be operated on by an operator. Operands may be as simple as numbers, or may be expressions. Generally operands that refer to the address of a data item are symbolic. The symbol is the label assigned to the data item by the programmer. If an operator requires multiple operands, they are simply listed after the operator--separated by commas.

Consider the following examples of actual 8080 statements:

<u>Statement</u>	<u>Type of Operand(s)</u>
ADI 16.	Literal Decimal Constant
STA NEWVALUE	Symbolic Reference to a Label
STA NEWVALUE+1	Symbolic Reference to the byte just 1 byte after the label "NEWVALUE"
MVI B,THRESHOLD/SCALE	The first operand is a symbolic reference to 8080 register B (see section for predefined symbols). The second operand is an expression, whose value will become the immediate byte value for the move immediate instruction.

ORIGINAL PAGE IS
OF POOR QUALITY

ASM80 will accept an expression for any operand. The value of the expression will be tested to determine that it is in the legal range of valid operands for the operator it is used with.

1.1.4 Comments

The programmer may piece notes about the program on every statement line by marking the beginning of the comment with a semicolon. The assembler will ignore any text after a semicolon. The programmer may use an entire line as a comment, by starting the line with a semicolon.

To make it convenient to enter large blocks of comments, a special type of comment called the block comment is allowed. All the text contained in the special delimiterpairs "/*" and "*/" will be ignored by the assembler. An example of their use is:

```
/* any text the programmer desires.....  
.....  
.....  
*/
```

Block comments have the additional property that they nest, thus the assembler will allow a block comment to contain another block comment. This allows the programmer to quickly block out a section of a program without worrying about enclosed comments. For example:

```
/* DELETING INITIALIZATION FOR TESTING
```

Statements

```
/*this section of code initializes the slot control que by
```

1) Scanning for

2) Initializing the map table....

...

...

ORIGINAL PAGE 1
OF FOUR QUALITY

*/

statements

END OF DELETION FOR TESTING */

In this example, the assembler recognizes the enclosed block comment as such, and does not terminate the enclosing comment prematurely.

1.2 SYMBOLS

The programmer may create symbols for use as labels or operands. A symbol consists of from one to 31 characters chosen from the following set:

The 26 letters	A,B,C,...,W,X,Y,Z
the 10 digits	0,1,2,3,...8,9
3 special characters	Dollar sign (\$), percent (%), and period(.)

The first character of a symbol may not be a digit. Examples of legal and illegal symbols are:

Legal

A
Al
\$PSTATUS
TOP.USED.CORE

Illegal

1A	(Digit illegal as 1st char)
A 1	(No spaces in symbols)
A&1	(& not in legal set)

If a symbol is longer than 31 characters, the excess characters are truncated, thus two symbols with the same initial 31 characters are indistinguishable to the assembler.

1.2.1 Special Reserved Symbols

A number of symbols have permanent values assigned to them by the assembler. These symbols include;

- 1) All 8080 instruction symbols,
- 2) All ASM80 psuedo-operators

For convenience in referring to the 8080 CPU registers, the following symbols have been predefined to select the corresponding register,

<u>Symbol</u>	<u>Register</u>
A	7-CPU accumulator
B	0-CPU register B
C	1-CPU register C
D	2
E	3
H	4
L	5-CPU register L
M	6-Specifies H,L pair
SP	In certain instructions, selects the stack pointer
PSW	In PUSH, POP instructions, selects the A register and Program Status Word.

In addition to the above symbols, the symbol "." is used to reference the current byte counter of the assembler. It may be used in an expression as any other symbol would be. At each occurrence, its value is the value of the byte counter at the start of the statement.

1.3 NUMBERS

Literal numbers may be used in statement to represent specific values. The assembler interprets a number in the source based on the current input radix and any local radix specifiers that may modify the number. A simple integer number may consist of characters from the following set:

the digits 0,1,2,...,8,9
six letters A,B,C,D,E,F

A number must begin with a digit. The simple integer number may be modified by a radix specifier drawn from the following set:

<u>Specifier</u>	<u>Radix</u>
.	Decimal
K	Octal (8)
O	Octal (8)
H	Hexidecimal (16)
X	Hexidecimal (16)
<u>R_n</u>	Base <u>n</u> , where <u>n</u> is in the range 2 to 16 inclusive. <u>n</u> is always given in the decimal base.

The specifier is appended to the number 1,3,

<u>Number</u>	<u>Value in Decimal</u>
10K	8
10X	16
-10R2	-2
0FX	15

By default, if no local radix specifier is given, the assembler assumes a number which is expressed in the decimal radix. The user may change the default by using the "IRDX" psuedo-operation. See Section 1.8.3 for details of this psuedo-op. In any case, a local radix specifier always overrides the default input radix.

ORIGINAL PAGE IS
OF POOR QUALITY

1.4 STRING TOKENS

1.5 EXPRESSIONS

An expression is a series of numbers and/or symbols separated by a special class of delimiters called arithmetic or logical operators. (We will simply refer to arithmetic or logical operators as "operators"--there should be no confusion with operators that are 8080 instructions.) The assembler evaluates the expression at assembly time by explicitly carrying out the operations indicated on the values of the numbers and symbols. The following operations are accepted by the assembler:

<u>Operations</u>	<u>Meaning</u>
NOT	logical complement
* , /	multiplication, Division
+ , -	addition, subtraction
&	logical and
! , XOR	logical or, logical exclusive

All arithmetic operations are signed integer 32 bit operations. All logical operations are also carried out over 32 bit values. This is possible since the assembler always maintains a 32 bit value for every symbol or number, even if that precision is not required for its particular occurrence. For example, labels are kept as 32 bit numbers, but in practice the value of a label must always fit in 16 bits since the 8080 can only handle memory addresses of 16 bits. At any point where the actual value must be truncated to fit the application, as an address field, the assembler checks to verify that the value is in the legal range for that application.

The order in which the assembler evaluates an expression is determined by the relative precedence of the operators in the expression. In the previous list of operators, the operators were listed in descending precedence, with those operators of equal precedence on the same line. Thus in evaluating the expression: $X = A+B*C$, the assembler will perform the multi-

plication first, then add A to the product of B and C.

If operators of equal precedence occur in the same expression, they are evaluated left to right. The following table of expressions demonstrates precedence evaluation.

<u>Expression</u>	<u>Value</u>
$5+3/2$	6
$5+2/3$	5
$5*2/3$	3
$5/2*3$	6
OFFX&1*2	2

For convenience, the assembler allows the use of parentheses to override the normal precedence evaluation of expressions. Any expression enclosed in parenthesis will be evaluated before applying the operation(s) just outside the parenthesis. For example, in the expression: $(A+B)*C$, the assembler will perform the addition first, since it must evaluate any enclosed expression before it "knows" the value to use in the multiplication operation.

1.6 ADDRESS ASSIGNMENTS

As source statements are processed, instruction and data bytes are assigned to consecutive memory addresses. The byte counter is incremented as required by a statement to assign its data bytes to memory. Some statements, such as assignment statements do not generate data and thus do not affect the byte counter.

1.6.1 Referencing the Byte Counter

The special symbol '.' (simple period) may be used in any expression to reference the value of the byte counter. The value of the byte counter used is the value for the first byte of the data generated by the statement, regardless of the number of bytes generated. Thus the symbol '.' has the value that would be assigned to a label, if one occurred on that line. For example,

"LABEL: JMP LABEL" and "JMP." are equivalent instructions.

ORIGINAL PAGE IS
OF POOR QUALITY

1.7 INSTRUCTION FORMATS

ORIGINAL PAGE IS
OF POOR QUALITY

1.8 ASM80 PSUEDO-OPERATORS

A special class of predefined psuedo-operators are used to control the operation of the assembler. The general format of these psuedo-operations is the same as for true operators.

1.8.1 END Statement

The END statement marks the end of the program for the assembler. Any program statements after an END statement are ignored by ASM80. An optional operand allows the programmer to specify the starting address of the program. For example:

END expression

The assembler evaluates the expression, and if binary output is being produced, an end block is generated with the expression value as the starting address of the program. The loader for the 8080 system will use this address to start the program executing. If the expression is missing, a value of -1 (FFFFH) is output for the starting address. By convention a -1 will stop the loader, but not start the program executing.

1.8.2 LOC Statement

The LOC statement allows the programmer to force the program byte counter to a new value. This allows altering the normal assignment of statements to the next sequential location. The LOC statement takes the form:

label: LOC expression : comment (Carriage return)

The expression is evaluated, the byte counter is set to the value. If the byte count is decreased as a result of the LOC statement a warning message is output, but it is accepted by the assembler. The value of any label on an LOC statement is the byte address that would have been assigned to an instruction statement or expression statement if one had been there. Thus the LOC statement

may be used to allocate tables, with the label taking on the value of the 1st byte in the table.

1.8.3 IRDX Statement

The IRDX statement allows the programmer to change the default input radix for conversion of numbers. The general form of the IRDX statement is:

IRDX expression ; comment (carriage return)

The expression is evaluated, and if the value is in the range 2 to 16 inclusive then the default input radix is set to the value. Note that the expression will be evaluated in the current input radix (before the IRDX is completely processed) so the user must use caution in specifying numbers in the expression. It is a good practice to always use a local radix specifier to assure you get the radix you desire. For example, "IRDX 8." will always set the input radix to octal regardless of the previous input radix.

1.8.4 ORDX Statement

The ORDX statement allows the programmer to change the default output radix of the assembler. Normally the assembler outputs program listings with the instruction locations and values in hexadecimal. The general form of the ORDX statement is:

ORDX expression ; comment (carriage return)

The expression is evaluated, and if the value is 2, 8, 10, or 16, then the output radix is changed to this new value. The listing will be formulated as required for the specified output radix.

ORIGINAL PAGE IS
OF POOR QUALITY

2.0 ASM80 OPERATION

ASM80 is designed to execute on a NOVA line computer system, under DOS or RDOS. It is assumed that the reader is generally familiar with the operation of the NOVA.

2.1 Creating the Source Program

Source program for the ASM80 can be created using the standard NOVA editor. As stated in earlier sections, each statement must be contained on a single line of source. A line may have as many as 132 characters, terminated by a carriage return.

It is often convenient to break a program up into pages organized by the logical function of the program. ASM80 accepts multipage programs as input, and if a listing is desired each page of input will be started on a new page of the listing file.

2.2 Executing the Assembler

ASM80 is invoked under DOS/RDOS as any other program would be; that is, the user simply types in the name, "ASM80" followed by the commands to the assembler. ASM80 will process the command line, then perform the indicated functions.

2.2.1 Basic ASM80 Command Line

The basic command, under DOS/RDOS, to invoke the assembler is:

ASM80 filename (carriage return)

This command will assemble the program filename. By default, no listing of the program will be produced. If any error occurs, the error messages will appear on the system console device (\$TTC). An absolute binary output file, in hexadecimal format, will be produced in the file "filename.B8". Any previous version of "filename.B8" will be deleted.

If a programmer desired, his program may be broken up into several files. In this case he should list the the filenames on the command line in the order they are

ORIGINAL PAGE IS
OF POOR QUALITY

to be read by ASM80. As an example consider assembling a 3 file program consisting of a parameter file, PARM: a standard system initialization file INIT; and the users program, SCANNER. The command line for this operation would be:

ASM80 PARM INIT SCANNER (carriage return)

This command will correctly assemble the program, but the binary output will go into a file called "BARM.B8". It is not likely that the programmer actually desires this, so the assembler accepts command line switches which modify its normal operation for the users convenience. Switches are discussed in the next section, but briefly the following command line would cause the binary output to go the file "SCANNER.B8" for the user.

ASM80 PARM INIT SCANNER SCANNER.B8/B (carriage return)

The item "/B" is called a local switch. It is local to the command line item "SCANNER.B8". ASM80 interprets a local B switch to specify the name of the output binary file.

2.2.2 ASM80 Command Line Switches

A switch is a command line item which modifies the interpretation of the command line by ASM80. An example of a command line with switches is:

ASM80/L F00 TESTF00/B (carriage return)

In this example, "/L" is a switch on ASM80 commanding that a listing of the program to be produced on the default listing device, "\$LPT". The "/L" is called a global switch since it modifies the command name. The switch "/B" is called local since it only modifies the processing of the filename is associated with in the command line, in this case "TESTF00".

The following table lists the global switches that may be used by the programmer:

<u>Switch</u>	<u>Function</u>
/K	Change the default listing radix from hexadecimal to octal. If not present, all assembler output of program addresses and program data is in hexadecimal. This switch is equivalent to giving a "OPDX 8." statement as the first statement of the program.
/L	By default no listing of the source program, and the corresponding binary output is given. If a global /L is specified, a program listing will be produced on the device "\$LPT".
/N	By default binary output is generated, If /N global is specified, no binary output will be generated. Normally ASM90 will generate the binary in a file created by appending ".B8" to the name of the first program file in the command line.

The following table lists the local switches that may be used:

<u>Switch</u>	<u>Function</u>
/E	If no listing file is being output, this switch specifies the name of an error logging file. All error messages will be dumped in this file. By default, if no listing is dumped and no /E is given, all error messages will be dumped on the system console (\$TTO). If a listing file is specified, all error messages will go to the listing file.
/L	This switch specifies that the filename it specifies will be used as the listing file. This allows listings to files other than the lineprinter.
/B	By default a binary output is produced in a file created by appending ".B8" to the filename of the first program file in the command line. If a /B is given, the filename it modifies will be used for dumping the binary output.

These switches are only those that are likely to be of use for the typical programmer. Appendix gives the details for other switches used in testing and debugging the assembler.

2.3 Assembler Output

ASM80 output falls into one of three classes, error messages, listings and binary.

2.3.1 Error Messages

When ASM80 detects an error in a program, an attempt is made to produce a reasonably specific error message to pin down the problem for the programmer. Error messages are dumped on the listing file if one was requested or on the error file if a /E was given and no listing requested. Finally, if no listing or error file was specified, the messages are dumped on the system console.

Error messages may be of three general classes:

<u>Type</u>	<u>Class</u>
E	Error that will almost certainly result in results other than those the programmer desired. Examples would be illegal register specification in an MOV instruction.
W	Warning of a possible error, but it is reasonable to expect the programmer might have intended this action. An example would be using this psuedo-op to back up the byte counter.
D	Diagnostic message issued by the assembler to aid in debugging the assembler itself. Users should never get any diagnostic messages.

In addition to these classes, some errors are of such a nature that it is impossible for the assembler to continue processing the program. In those cases the error is called Fatal, and will be indicated as such in the error message. An example of a fatal error would be running out of symbol

ORIGINAL PAGE IS
OF POOR QUALITY

table space.

The general format of error messages is:

level nnnn error message text

FATAL level nnnn text

where level is one of the three level specifier characters (E,W,D), nnnn is the decimal error code assigned to this error message. If an error is fatal, it will be preceded with the work "FATAL".

In the listing file, error messages precede the line of source they apply to. In the error file, the line of source on which the error was detected is output also.

In some cases, the error handler has a reasonable chance of identifying the actual source symbol that is primarily responsible for the error. In those cases, the symbol is included in the error message, enclosed in double quotation marks. In some cases, the enclosed symbol will be a blank--this generally means that the assembler was expecting a delimiter at this point other than a blank, i.e., an operator in an expression.

Appendix _____ contains a full listing of all error messages that may be output by the assembler.

2.3.2 Listing Output

If requested by the presence of a /L switch, the assembler will produce a listing of the source program. This listing follows one of three formats

depending on the source line statement:

<u>statement</u>	OR
<u>iijjkkll statement</u>	or
<u>nnnn iijjkkll statement</u>	

In the first example, the statement required no processing by the assembler, i.e., a comment, thus the statement was simply reproduced in the listing. In the second case, the assembler had to evaluate an expression, but the statement did not generate binary output, so the value of the expression is listed, with a precision of up to 4 bytes. ii, jj, kk, and ll each represent a single byte in hexadecimal notation. Finally for the case of a statement that produces binary code, the byte address of the code is given by nnnn in hexadecimal, followed by the value up to four bytes in length.

While the format will always follow this general organization, the precise field widths will be adjusted to reflect the current output radix of the assembler. For example if octal output is selected the location field, nnnn requires 6 octal digits to represent any possible Intel address.

2.3.3 Binary Output

By default binary output is created for any program assembled. This binary consists of a series of data blocks that specify the data to be loaded into the Intel and the address to load it. After

all the data blocks, an end block indicates the starting address of the program (see END psuedo-op). Normally the binary will be produced on hexadecimal format, suitable for transmission over system that can only process ASCII characters. For full details of the Binary format see Appendix ____.

SIGNAL PREPROCESSING AS AN AID TO
ON-LINE EKG ANALYSIS

by

Joseph Bennett Walters, Jr.

Submitted in Partial Fulfillment
of the Requirements for the
Degree of Bachelor of Science
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June, 1971

Signature of Author

Department of Electrical Engineering

Certified by.

St. Burrage Flegenbaum

Thesis Supervisor

Accepted by.

Robert Z. Kyhl

Chairman, Departmental Committee on Theses

PRECEDING PAGE BLANK NOT FILMED

ABSTRACT

The problems of conventional digital sampling are discussed in the context of patient monitoring and long term data collection. Alternative methods of data compression are presented.

The capability of three redundancy reduction algorithms to produce adequate representations of electrocardiographic data was examined. It was found that a zero order interpolator was the best redundancy reduction algorithm in terms of data reduction capability and overall quality of the reconstructed EKG.

A hardware device was constructed which carried out zero order interpolation on a signal. Several alternative methods of implementing a zero order interpolator were analyzed in terms of the demand they placed on a small general purpose computer. It was found that the computer could handle a moderate number of channels even with the worst implementation, while the best implementation produced negligible demands on the computer.

Examination of the zero order interpolators reconstructed signal indicated that this representation was adequate for analysis of rhythm.

ACKNOWLEDGEMENT

**I wish to thank my thesis advisor, Prof. S. K. Burns,
for his guidance, assistance, and patience in this work.**

TABLE OF CONTENTS

Abstract	2
Acknowledgement	3
List of Figures and Tables	5
Chapter I - Introduction	
Normal Heart Function	7
Abnormal Heart Function	13
Clinical Significance of Ectopic Beats	19
Patient Monitoring	20
Longterm Data Collection	22
Conventional Digital Signal Representation	23
Alternative Approaches to the Representation of EKGs	28
Chapter II - Redundancy Reduction	
Introduction	30
Predictors and Interpolators	34
Zero Order Predictor	35
First Order Predictor	38
Zero Order Interpolator	41
Results of Experimental Runs	45
Chapter III - Alternative Implementations of the Zero Order Interpolator	
Hardware Implementation	60
Processor Overhead for Alternative Implementations of the Zero Order Interpolator	68
Chapter IV - Conclusions	
General	75
Appendix	
General Information	77
System Control Logic	80
Analog Design	85
Bibliography	100

LIST OF FIGURES AND TABLES

Chapter I

Figure 1 -	Conduction System of the Heart-----	8
Figure 2 -	Excitation and Contraction of the Heart During Normal Sinus Rhythm-----	11
Figure 3 -	Excitation and Contraction of the Heart During a VPB-----	14
Figure 4 -	Examples of a VPB, APB, and Nodal Beat-----	16
Figure 5 -	Sampling Ratios Required for a Given Accuracy, as a Function of Roll-off Rate--	24

Chapter II

Figure 6 -	Categories of Data Compression-----	31
Figure 7 -	Operation of a Zero Order Predictor-----	36
Figure 8 -	Operation of a First Order Predictor-----	39
Figure 9 -	Operation of a Zero Order Interpolator-----	43
Figure 10 -	Plot of Reduction Ratio vs. % RMS Error---	46
Figure 11 -	Reconstructed ZOI, ZOP, and FOP Output for K=10-----	49
Figure 12 -	Reconstructed ZOI, ZOP, and FOP Output for K = 20-----	51
Figure 13 -	Reconstructed ZOI, ZOP, and FOP Output for K = 30-----	52
Figure 14 -	Reconstructed ZOI, ZOP, and FOP Output for K = 40-----	53
Figure 15 -	Reconstructed ZOI, ZOP, and FOP Output for k = 50-----	54
Figure 16 -	Reconstructed ZOI, ZOP, and FOP Output for K = 60-----	55
Figure 17 -	Reconstructed ZOI, ZOP, and FOP Output for K = 70-----	56
Figure 18 -	Reconstructed ZOI, ZOP, and FOP Output for K = 80-----	57
Figure 19 -	Reconstructed ZOI, ZOP, and FOP Output for K = 90-----	58
Figure 20 -	Reconstructed ZOI, ZOP, and FOP Output for K = 100-----	59

Chapter III

Figure 21 -	Block Diagram of ZOI Implementation-----	61
Figure 22 -	Basic Hardware Timing Diagram-----	64
Figure 23 -	Example of Direct Output of the Hardware--	66
Figure 24 -	Percentage of CPU Time vs. Number of Active Channels-----	73

Appendix

Figure A1 - Block Diagram of ZOI Hardware-----	79
Figure A2 - System Control Logic-----	81
Figure A3 - System Timing Diagram-----	82
Figure A4 - Signal Conditioning Stage-----	86
Figure A5 - Sample and Hold-----	87
Figure A6 - Maximum Tracker-----	90
Figure A7 - Unity Gain Inverter-----	92
Figure A8 - Voltage Averager Stage-----	94
Figure A9 - Aperture Width Comparator-----	96
Figure A10 - Examples of Direct System Output-----	99
 Table 1 - Software Event Timing -----	 69

CHAPTER I

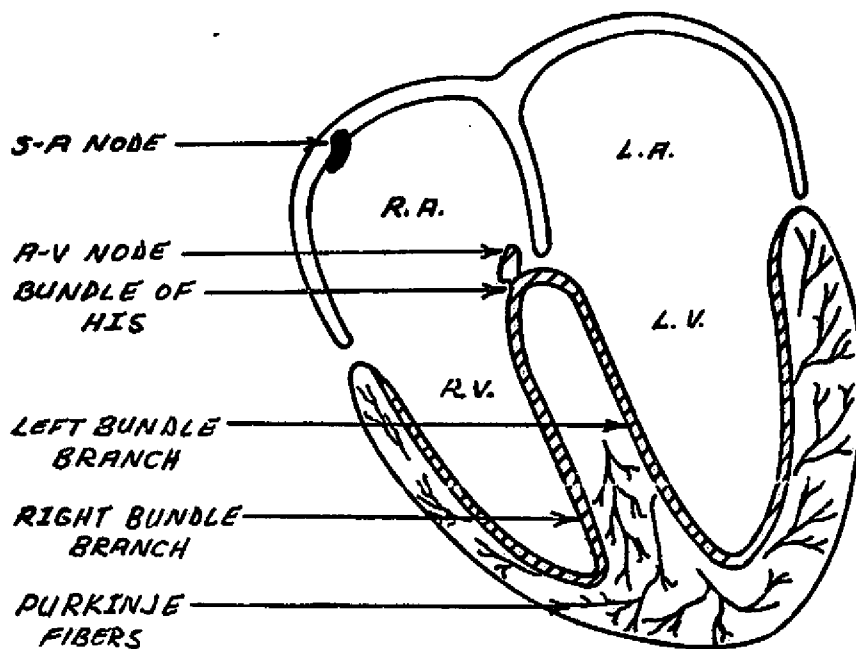
INTRODUCTION

Normal Heart Function(2):

The normal heart functions as two mechanically independent pumps. The right atrium (Figure 1) and ventricle pump blood through the lungs while the left atrium and ventricle pump blood into the systemic circulation. The mechanical action of the heart is coordinated by specialized tissue which forms the conduction system. This conduction system is rhythmically activated by a pacemaker.

The normal pacemaker is the sino-atrial (S-A) node (Figure 1). The S-A node initiates atrial excitation. As this excitation spreads, the atrial muscle is depolarized and contracts. When the atrial excitation reaches the atrioventricular (A-V) node (the only electrical connection between the atria and ventricles) it is delayed slightly before entering the Bundle of His. The Bundle of His bifurcates into the left and right branch, distributing the excitation to the Purkinje fibers in each ventricle. The Purkinje fibers interlace the ventricular muscle, distributing the excitation throughout the ventricles. The propagation velocity of the excitation through the Bundle of His (~ 4000 mm/second) is about an order of magnitude greater than the propagation velocity through the muscular tissue, thus all muscular activity is effectively synchronized.

Figure 1 - Conduction System of the Heart. Drawing of the conduction system of the heart demonstrating the S-A node, A-V node, bundle of His, and the Purkinje system. After Corday and Irving (2).



ORIGINAL PAGE IS
OF POOR QUALITY

Muscular activity of the heart is reflected in the electrocardiogram, or EKG. The EKG is a recording of potential differences, vs time, between points on the subjects skin. Atrial depolarization (Figure 2) produces the P wave. While the amplitude of the P wave will vary with the position of electrodes on the skin, the width is less than 0.10 seconds in the normal subject. Ventricular depolarization is reflected in the QRS complex. The width of the QRS complex is also less than 0.10 seconds. Repolarization of the ventricles produces the T wave. The Ta wave, generated by repolarization of the atria, is usually hidden by the QRS complex. The complete P-QRS-T complex occurs over about 300 msec.

Figure 2 - Excitation and Contraction of the Heart During Normal Sinus Rhythm. The darkened area of the heart indicates the area the excitation spreads through to produce the darkened section of the EKG immediatly below. (the S-A node is always darkened) The first small deflection produced (a-Atrial Excitation) is the P wave. This is followed by the large QRS deflection during ventricular excitation (d). Finially the T wave is produced by ventricular repolarization during ventricular systole(e). After Corday and Irving (2).

a) ATRIAL
EXCITATION



b) ATRIAL
SYSTOLE



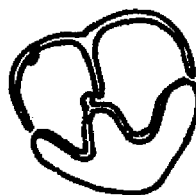
c) ATRIAL
DIASTOLE



d) VENTRICULAR
EXCITATION



e) VENTRICULAR
SYSTOLE



f) DIASTOLE



ORIGINAL PAGE IS
OF POOR QUALITY

**INTRODUCTORY SUMERIAN
CHRESTOMATHY**

Thorkild Jacobsen

Abnormal Heart Function(2):

Abnormalities of the heart may be classified as purely mechanical as well as electrical. Electrical abnormalities often reflect structural damage; however, they may occur with no apparent mechanical damage. Abnormalities in the electrical activity of the heart may be classified as either disturbances of pacemaker function or conduction defects.

Disturbances of pacemaker function manifest themselves as disturbances of the S-A node or 'capture' of the pacemaker function by tissues other than the S-A node. An area which captures the pacemaker function (other than the S-A node) is called an ectopic focus. Since an ectopic focus initiates the excitation of the heart from an abnormal location, the sequence of events in the heart cycle may be significantly altered greatly reducing the efficiency of the heart. Figure 3 schematizes the sequence of events for one possible ventricular ectopic beat. Such beats usually come earlier than expected for a normal beat, and are called ventricular premature beats (VPBs). Figure 4a provides a stripchart recording of a VPB. The QRS complex for the VPB is premature and widened, due to the aberrant conduction of the ventricular excitation. The specialized conduction system does not conduct the excitation as it would for a normal beat.

The ectopic focus may be located in the atria also, in which case the beat is called an atrial ectopic beat. The excitation spreads abnormally through the atria resulting in

Figure 3 - Excitation and Contraction of the Heart During a VPB. The darkened area of the heart indicates the area of the heart the excitation is spreading through to produce the darkend section of the EKG below each heart. The first figure represents diastole of the previous (normal) beat. After Corday and Irving (2).

DIASTOLE

ECTOPIC VENT.
EXCITATIONVENTRICULAR
SYSTOLERETROGRADE
ATRIAL EXCIT.ATRIAL
SYSTOLE

DIASTOLE



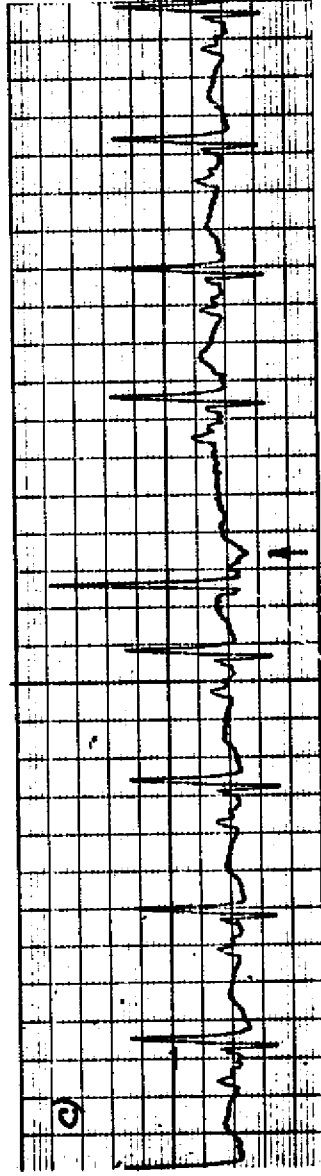
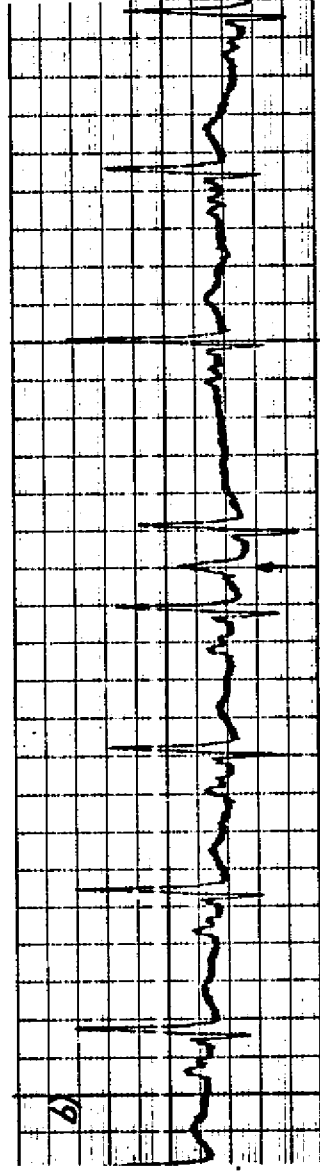
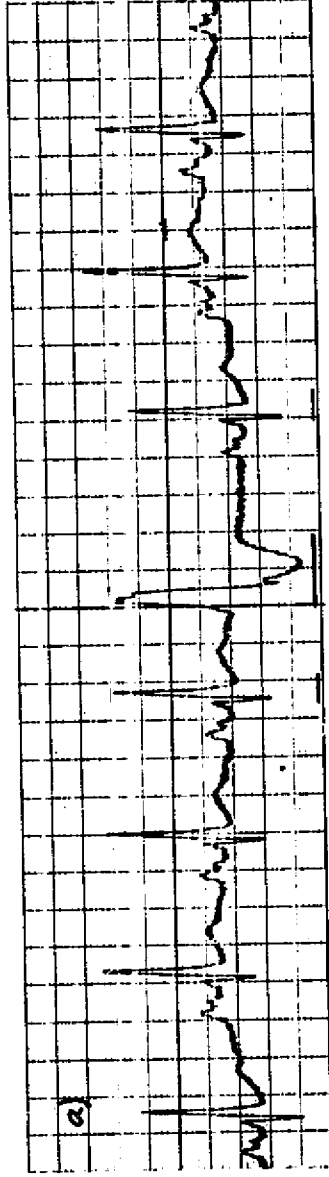
ORIGINAL PAGE IS
OF POOR QUALITY.

Figure 4 - Examples of a VPB, APB, and Nodal Beat. All these recordings came from the same patient. Note that his 'normal' P wave is abberantly conducted, resulting in a double peak.

a) A VPB. The QRS complex is widdened, permature, and followed by a compensatory pause.

b) A APB. Again the ectopic beat is premature. The QRS is normal, but preceeded by an abnormal P wave.

c) A nodal premature beat. The QRS is normal, but followed by an inverted P wave.



an abnormal P wave morphology. After the excitation^z is passed to the A-V node, the ventricular excitation sequence is normal, resulting in a normal QRS morphology. Atrial ectopic beats are also premature and are referred to as atrial premature beats (APBs). Figure 4b provides an example of an APB.

If the ectopic focus is in or near the A-V node, or the Bundle of His before the bifurcation, the resultant beat is termed a nodal premature beat. Atrial excitation progresses in a retrograde fashion, thus if a P wave is visible, it is inverted. The P wave may be buried in the QRS complex, which is essentially normal in shape. Figure 4c provides an example of a nodal premature beat.

Conduction defects alter the normal passage of the excitation through the heart. The changes may take the form of a block, in which the conduction process is interrupted, or abnormal pathways. A block in the S-A node will result in failure of the heart to be paced by the S-A node; another focus must take over the pacing function. Typically the atrial tissue is most likely to take over the pacing function in the absence of S-A node initiation. A block at the A-V node prevents normal passage of the excitation from the atria to the ventricles. Ventricular activity may now originate in the A-V node or in the ventricles themselves. However, independent of its origin, ventricular activity is now unrelated to atrial activity.

Clinical Significance of Ectopic Beats(2):

The clinical significance of the rythm defects mentioned in the previous section is not fully understood. VPBs are known to occure in otherwise normal subjects. Unless their severity is sufficient to cause patient distress (one can often feel the pause after a VPB), nothing is usually done about them. The exact incidence of VPBs in normal subjects is not known. In the cardiac patient who is recovering from a heart attack, or other tramatic events involving the heart, they are suspected^t of being the first warning of impending heart failure through progressively more serious arrhythmias, such as ventricular tachycardia and ventricular fibrillation. The frequency and variety (number of different foci) of VPBs is generally considered an indication of the irritability of the heart and influences decisions concerning the nature of the treatment given the patient.

Patient Monitoring:

Since VPBs, and other ectopic rhythms, are considered a measure of the irritability of a cardiac patients heart, their detection by a monitor is of prime importance. A monitor for cardiac patients should meet the following requirements:

- 1) It should be able to detect ectopic beats in real-time and provide some record of their occurrence.
- 2) It should be operable even if the patient's 'normal' rhythm, or beat morphology, would be considered abnormal.
- 3) It should be insensitive to muscle noise, pacemaker and other artifacts.
- 4) It may require training on a sample of the patients normal rhythm, but should not require training on ectopic beats.
- 5) It should be as economical as is consistent with the above requirements.

The first requirement, 1, is the most basic requirement of the monitor. The monitor must be able to monitor a patient continuously in real-time. Periodic monitoring of the patient offers little advantage over the current practice of intensive care units, where a nurse periodically monitors the patient. The ideal record of the occurrence of an ectopic event would be a stripchart recording of the event and a few seconds of data around it. This record is in a form which the attending staff is familiar. Long term trend records would also be useful.

Proper operation during a stable arrhythmia, requirement 2, recognizes the fact that any patient in a cardiac care unit, in all probability, has something wrong with his heart.

Thus his 'normal' rhythm may in fact be abnormal for the general population.

Recognition of various artifacts that may occur in the EKG, requirement 3, is important. If possible the monitor should extract the EKG signal from the noise, but when this is not possible, it should be able to shut itself off, thus preventing false alarms. Excessive false alarms will result in inattention to the monitor, a condition which must be avoided.

The monitor must be able to detect previously unseen events, hence requirement 4 demands that only the patient's normal rhythm be necessary for the establishment of detection criteria.

Finally, requirement 5 recognizes the rising costs of medical care in the United States.

The author feels that the best approach to meeting these requirements, at the present time, is through the small general purpose computer. This is especially true at the research level. In Chapters II and III the author will examine one problem in the application of computers to the patient monitoring problem.

Longterm Data Collection:

As pointed out in the section on the clinical significance of ectopic beats, the true incidence of ectopic beats in normal subjects is not known. One way to obtain this information is to record the EKG of a large population of subjects during normal daily activity, then analyze the records. Say that one wishes to monitor a subject during normal daily activity for eight hours. If we assume that the important components of the EKG (in so far as the detection of ectopic beats is concerned) lie below 50 Hz, and filter the waveform so that it is bandlimited to 50 Hz; then a sampling rate of 100 Hz is required according to the Nyquist criterion (9). An eight bit sample provides adequate accuracy for this purpose, thus 2.3×10^7 bits are required to record eight hours of EKG. There is simply no practical way to record this much data without severely restricting the subjects mobility. Even if the data can be recorded, what does one do with it? Without some form of automated analysis, the systematic study of such large amounts of data is impossible at any reasonable cost.

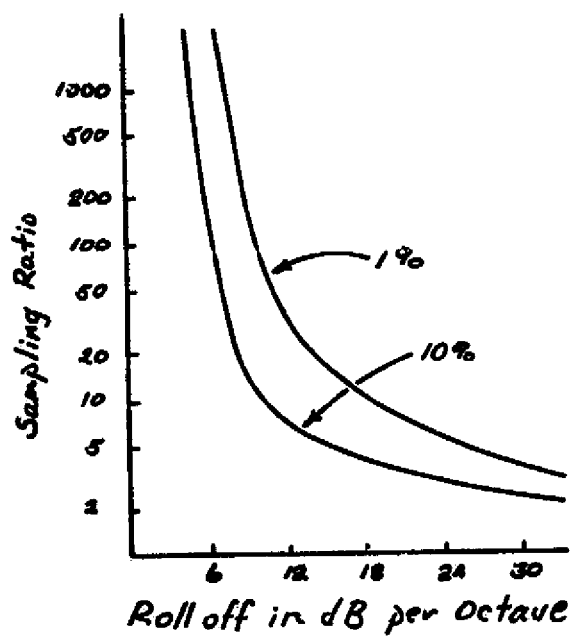
It is apparent that a monitor meeting the requirements of the previous section would be suitable for processing such longterm records. If possible, it should operate faster than real-time to expedite the analysis of the records.

Conventional Digital Signal Representaion (5, 9, 10):

The conventional approach to digital representation of signals is uniform sampling of the source signal and subsequent conversion of these samples into digital representation. The Nyquist criterion states that the sampling rate must be at least two times the period of the highest frequency component present in the source signal. If this condition is met, the source signal may be recovered by lowpassing a uniformly spaced impulse train with amplitudes specified by the sample values.

It should be noted that the Nyquist rate is determined by the highest frequency component present, not the highest one of interest. The usual technique is to reduce the bandwidth to the region of interest by lowpassing the signal, then sampling the lowpassed signal at an appropriate rate. In practice this does not produce truly bandlimited signals, since all practical filters pass some components above their 'cutoff' frequency. Figure 5, after Macy(10), indicates the actual sampling rates required for a specified accuracy, given a variety of filter roll-off rates and a non-bandlimited source signal. For this figure the input is assumed to have a flat spectrum, and while this is not generally true in signals of interest, the figure is useful in that it indicates the magnitude of error one might expect. The possible error is quite large, and samples must be collected at higher rates than indicated by applying the Nyquist criterion to the cutoff frequency of the filter.

Figure 5 - Sampling Ratios Required For a Given Accuracy, As a Function of Roll-off Rate. The sampling ratio is (sampling frequency/filter cut-off frequency). The errors are computed only for aliasing effects, errors due to nonoptimal reconstruction are not included. After Macy (10).



ORIGINAL PAGE IS
OF POOR QUALITY

While it might seem wise to have as sharp a cutoff as is possible, this may not be the case. In analog filters, caution must be used when 'simply' increasing the roll-off rate of a filter. It should be recalled that the phase characteristics of the filter are being altered. Poor phase characteristics will distort the data. In fact, the time necessary for the transient portion of the filter response to decay is inversely related to the 'sharpness' of the roll-off of a low pass filter.

The selection of a sampling rate also depends on the nature of the data processing to be carried out on the digital representation of the signal. If the processing is to be carried out on a reconstructed signal, e.g. by digital lowpassing, then the signal may be reconstructed to an arbitrary degree of accuracy (limited only by the resolution offered by the number of bits used for the sample representation). The errors presented in Figure 5 assume that optimal (lowpassing) reconstruction was used on the sample points. If optimal reconstruction is not used the error will be even greater.

If the application requires that the sampled data be presented for human observation, an additional constraint on the sampling rate must be applied. Human observers tend to interpolate data by connecting geometrically nearest neighbors. Thus a 59 Hz signal, sampled at 120 samples per second, appears to the observer as two 1 Hz sinusoids in opposite phase. To guarantee that consecutive samples are, in fact, geometric nearest neigh-

bors requires a rather high sampling rate; something on the order of 10 times the highest frequency component in the signal.

Alternative Approaches to the Representation of EKGs:

The large amount of data that must be collected and stored for subsequent automated analysis of EKGs has resulted in interest in signal representation methods other than conventional sampling. Ideally these methods would be more efficient than sampling and easier to process. They must be practical to implement also.

Young and Huggins (//) have represented the EKG with a sum of 12 empirically defined orthonormal exponentials. This representation is very compact, requiring only 12 values for each EKG complex (QRS and T wave only, they have not included the P wave in the representation). Using these 12 exponentials they were able to represent a set of 18 EKG complexes with an error of only 5%. Their selection of exponentials was empirical and they make no claim that the set they have chosen can represent any possible EKG complex to the same accuracy.

The diagnostic capability of this representation has been studied(/2). Young and Huggins found that in trials with about 50 complexes, they were able to separate the data into 5 disease classes and normals with about 70% accuracy.

Cox et.al. (3) have proposed another technique for representing the EKG. They call their method AZTEC, for Amplitude Zone Time Epoch Coding. AZTEC reduces the number of bits required to describe an EKG by specifying the waveform in terms of bounds and slopes rather than sample points. Bounds are specified as amplitudes and lengths (or durations). A bound

provides the information that the input waveform was within a specified range of the bound amplitude for the bound length. During the rapid action of the QRS complex, many short bounds are produced. If a string of short bounds ($< 8\text{msec.}$) is encountered, and there has been no slope change in the input waveform, the string of bounds is replaced by a slope segment. The slope segment specifies the waveform by a slope value and length. The input waveform may be reconstructed by interpolating a straight line of specified duration and slope from the amplitude specified by the preceeding bound. Cox et. al. (3) report that AZTEC is capable of providing a 10 to 1 reduction in the number of bits required to describe a typical EKG waveform.

CHAPTER II

REDUNDANCY REDUCTION

Introduction:

Figure 6, after Kortman(8), depicts various categories of data compression. All of these methods attempt to reduce the bandwidth required to transmit a signal, that is the number of bits per unit time required to describe the signal to a given accuracy. These methods are parameter extraction, adaptive sampling, redundancy reduction, and encoding.

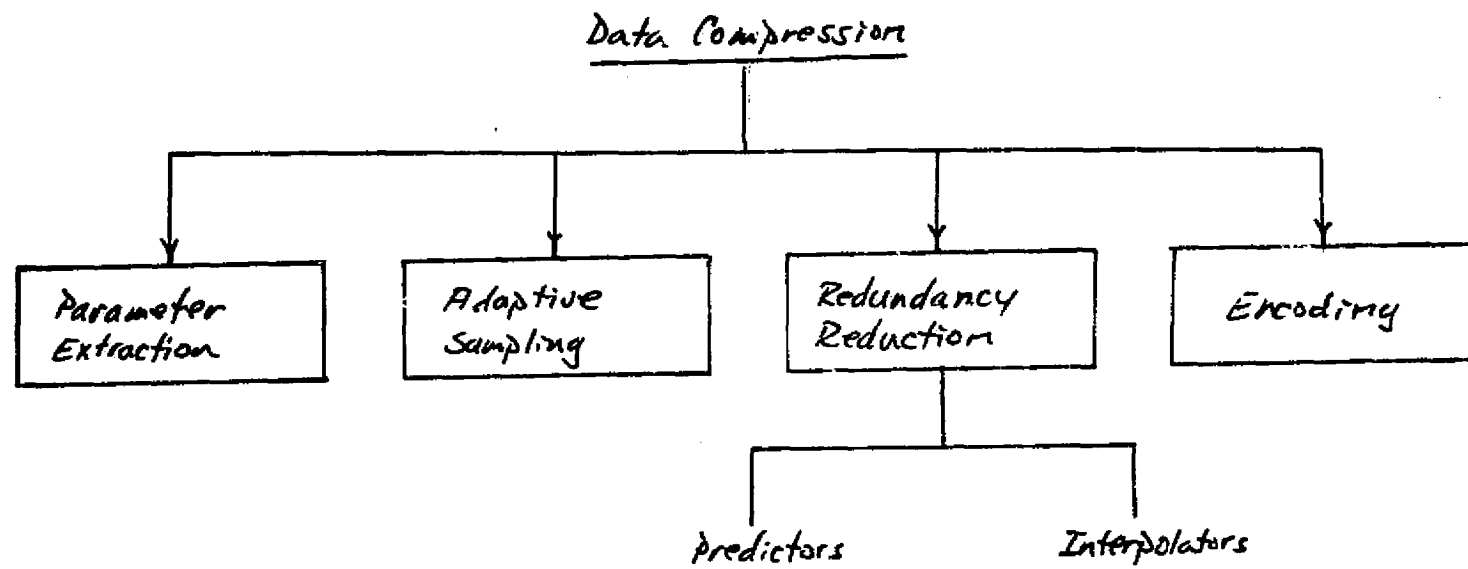
Parameter extraction (8) reduces the number of bits required to describe a waveform by extracting the significant parameters of the signal and transmitting only the parameters. Thus if one were interested in the detection of VPBs one might construct hardware to extract the width of the QRS complex and the interval between complexes.

Parameter extraction depends on a priori knowledge of the signal and a detailed knowledge of the purpose of the analysis to be preformed. In some cases the device to extract the parameters may be more complex than the analysis hardware.

Adaptive sampling (8) provides bandwidth compression by adjusting the sampling rate of the signal to match the required bandwidth. Thus samples would be transmitted only when there is activity in the signal.

Typically one is interested in ⁴garanting that the digital representation of the signal is accurate to a given error, rather than insist that it can be recovered perfectly. In

Figure 6 - Categories of Data Compression. This figure illustrates various methods of data compression . After Kortman(8) .



this case, redundancy reduction techniques (1,6,8) are useful. These techniques remove redundant information from the signal and transmit only that information required to represent the signal to a specified accuracy.

Encoding (8) includes such techniques as delta modulation and incremental encoding. An incremental encoder transmits the difference between adjacent samples rather than the sample values. By adding the transmitted difference into a register (integrating the transmission values) one produces a representation of the signal at the receiver. Delta modulation is essentially a form of incremental encoding in which the transmitted values are limited to one bit. The receiver will add one unit to the register if a 1 is received and subtract a one if a zero is received.

Predictors and Interpolators:

Redundancy reduction techniques fall into two general classes (1, 8) , predictors and interpolators.

Predictors make a predictions about the future behavior of the signal and transmit the parameters of the prediction. As the source signal varies in time, the actual value is compared with the predicted value. When the signal has deviated from the prediction by more than a specified amount, a new prediction is generated and transmitted. We will consider two types of predictors later in this chapter; the zero order predictor and the first order predictor.

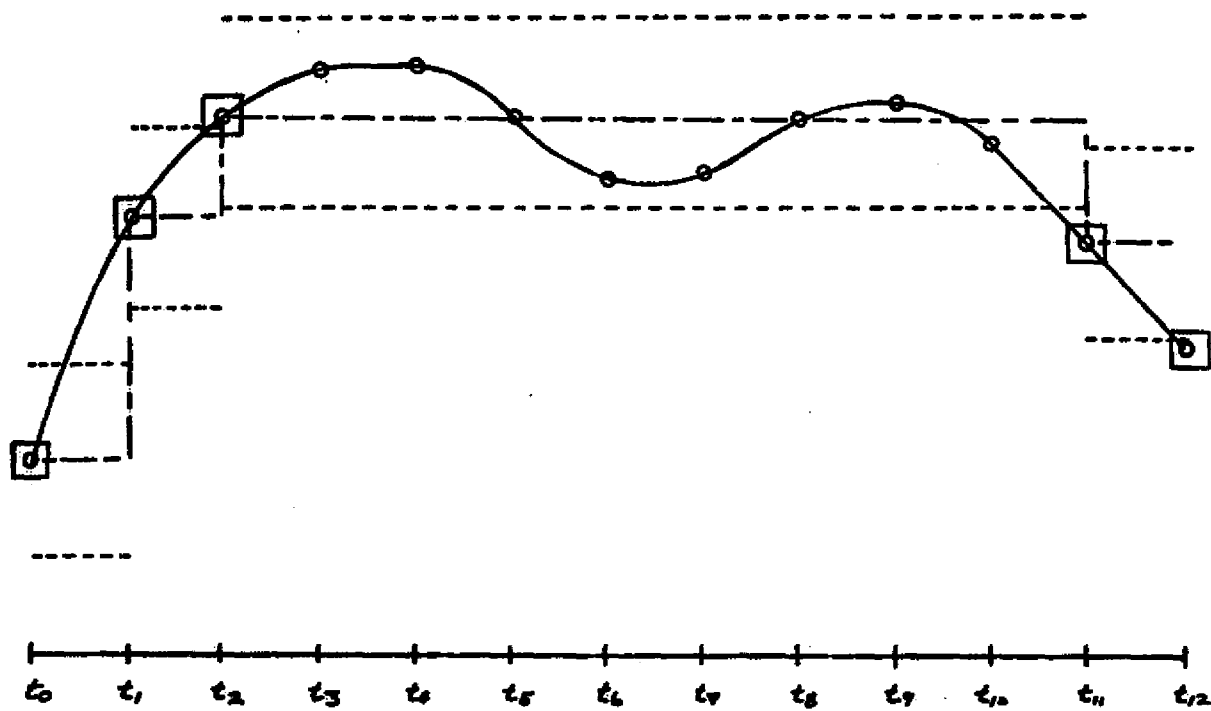
Interpolators generate a representation of a source signal based on the past behavior of the signal rather than predictions about its future behavior. Generally interpolators use some heuristic method for obtaining a good representation of the history of the signal. They may of course use an optimal technique, such as linear regression (if one is interested in representing the data with a line); but these methods are more difficult to implement. We will be concerned with the zero order interpolator, which is equivalent to the generation of bounds by AZTEC (3).

Zero Order Predictor :

The zero order predictor (ZOP) predicts that the waveform will remain constant in the future. Figure 7 depicts the operation of a ZOP. Consider the initial sample of the first segment in Figure 7, at time t_0 . The ZOP transmits this sample value as its prediction and sets up an aperture of width $2K$ centered on the transmitted value. This aperture defines the range the signal may occupy before a new prediction will be made, thus we always know the value of the source signal to within $\pm K$. New predictions are transmitted at t_1 and t_2 since the signal left the previous aperture range on both sample points. The signal does not leave the aperture centered on the t_2 sample until time t_{11} . In this example, five values are transmitted while 12 samples were required.

Under worst case conditions, note that the device would transmit every sample point, and thus become a conventional sampling system.

Figure 7 - Operation of a Zero order Predictor. The solid line represents the input signal, which is sampled uniformly in time at the points indicated by the circles. The squares indicate the transmitted samples. The reconstructed signal is represented by the line of alternating short and long dashes. Aperture boundaries are indicated by the dotted lines.



First Order Predictor:

The first order predictor (FOP) (1) approximates a signal with a line of specified starting amplitude, slope and length. Figure 8 depicts the operation of a FOP. Consider the sample at t_0 to be the initial sample of a new approximation to be generated by the FOP. The FOP waits for an additional sample at t_1 , and then predicts that the signal can be represented by a line starting at the sample value of t_0 , with a slope specified by the difference between the sample values of t_0 and t_1 . Apertures of $2K$ are centered on each predicted point, and the FOP checks to see if the signal sample value falls within the aperture. As can be seen, the prediction fails for the sample value at t_2 , and a new prediction must be made.

Note that the transmission of a prediction requires the transmission of two sample values, rather than one as required in the ZOP. One might expect the FOP to be very sensitive to noise on the signal, since it bases the prediction on the derivative of the signal. In the worst case, the FOP will limit out to a sample system, since it must effectively transmit every sample point.

Figure 8 - Operation of a First Order Predictor. The solid line represents the input signal which is uniformly sampled in time at the points indicated by the circles. The squares indicate the values that must be transmitted. The reconstructed signal is indicated by the alternating short and long dashes.

Zero Order Interpolator:

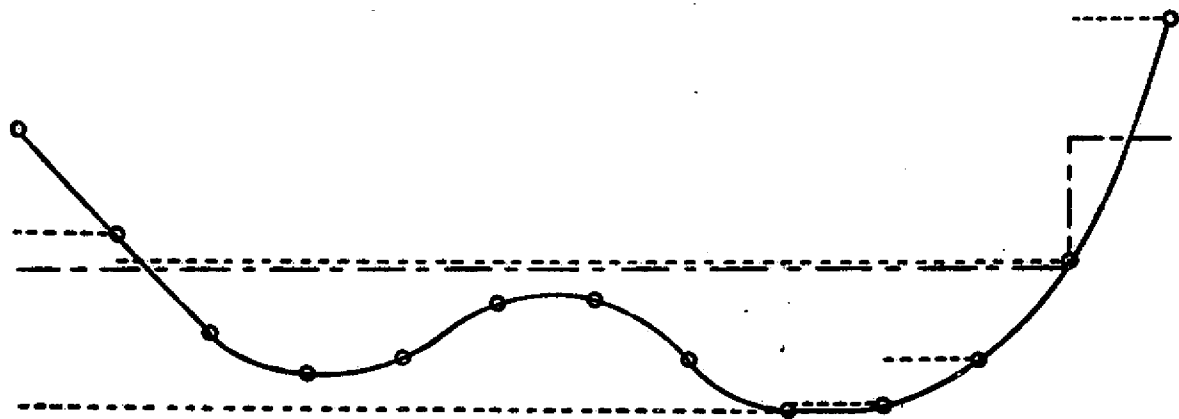
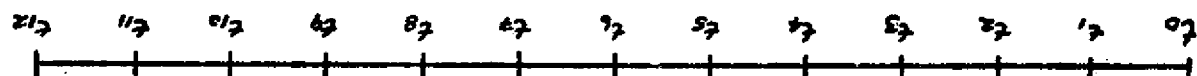
The zero order interpolator (ZOI) (6,8) may be considered to be a heuristic for maximizing the amount of time the source signal remains within a floating aperture. Figure 9 depicts the operation of a ZOI. Consider time t_1 to be the terminating point for a previous segment of the approximation. We define a V_{\max} , the maximum value of the source signal since the start of a segment, and V_{\min} , the minimum value of the source signal. V_{\max} and V_{\min} define the current boundaries of the floating aperture. Note, in Figure 9, that the aperture expands as samples for t_2 through t_{11} are taken in. When the sample at t_{12} is taken in, the aperture expands beyond the maximum allowed width, K . An interpolated value is now transmitted. In the example of Figure 9 and in the results described later in this chapter, the interpolated value is the center of the aperture at time t_{12} , the time at which the error was discovered. An alternative implementation of the ZOI, hereafter referred to as a bounded ZOI, generates the interpolated value from the center of the aperture at t_{11} , the period previous to the detection of the out of bounds condition. In this case the errors of the ZOI are always bounded to $\pm 0.5K$. The author chose the simpler, unbounded implementation since it does not require storage of the previous aperture center.

In the worst case, that is the difference between sample values exceeds K all the time, the unbounded system transmits half as many samples as there are samples. Each transmitted

PRECEDING PAGE BLANK NOT FILMED

value is the average of two adjacent signal samples, thus the input signal is being digitally lowpassed. In the case of the bounded ZOI the device transmits every sample point, and behaves as a conventional sampling system.

Figure 9 - Operation of a Zero Order Interpolator. The solid line indicates the input signal, which is sampled uniformly in time at the points indicated by the circles. The dashed lines indicate the aperture boundaries. The reconstructed signal is indicated by the line of alternating short and long dashes. The squares indicate the transmitted values. Note that the reconstructed signal could not have been generated in real-time since the transmitted values occur at the end of each segment. A device with a memory is required to reconstruct the signal. Since this memory must be bounded, the maximum length approximating segment is also bounded.



ORIGINAL PAGE IS
OF POOR QUALITY

Results of Experimental Runs:

In order to compare the ZOP, FCP, and ZOI on EKG data, the author has executed each algorithm on typical EKG data (channel 5, PBBH-5 tape). Each algorithm was programed for the NOVA computer (4) in the BASIC programming language (7). Each algorithm was executed on exactly the same set of signal samples. The results are compared in Figure 10, a plot of experimental reduction ratio vs. per cent RMS error. The reduction ratio is defined as,

$$\frac{\text{number of output segments}}{\text{number of input samples}} .$$

The reduction ratio does not measure the actual reduction in bandwidth required to transmit the representation.

The per cent RMS error is defined as follows; let the source signal values be represented by V_i^S , where i runs from 1 to N , the total number of samples. The reconstructed signal values for each sample point are given by V_i^R , where i again runs from 1 to N . Then the RMS Signal Value is given by,

$$\sqrt{\frac{\sum_{i=1}^N (V_i^S)^2}{N}}$$

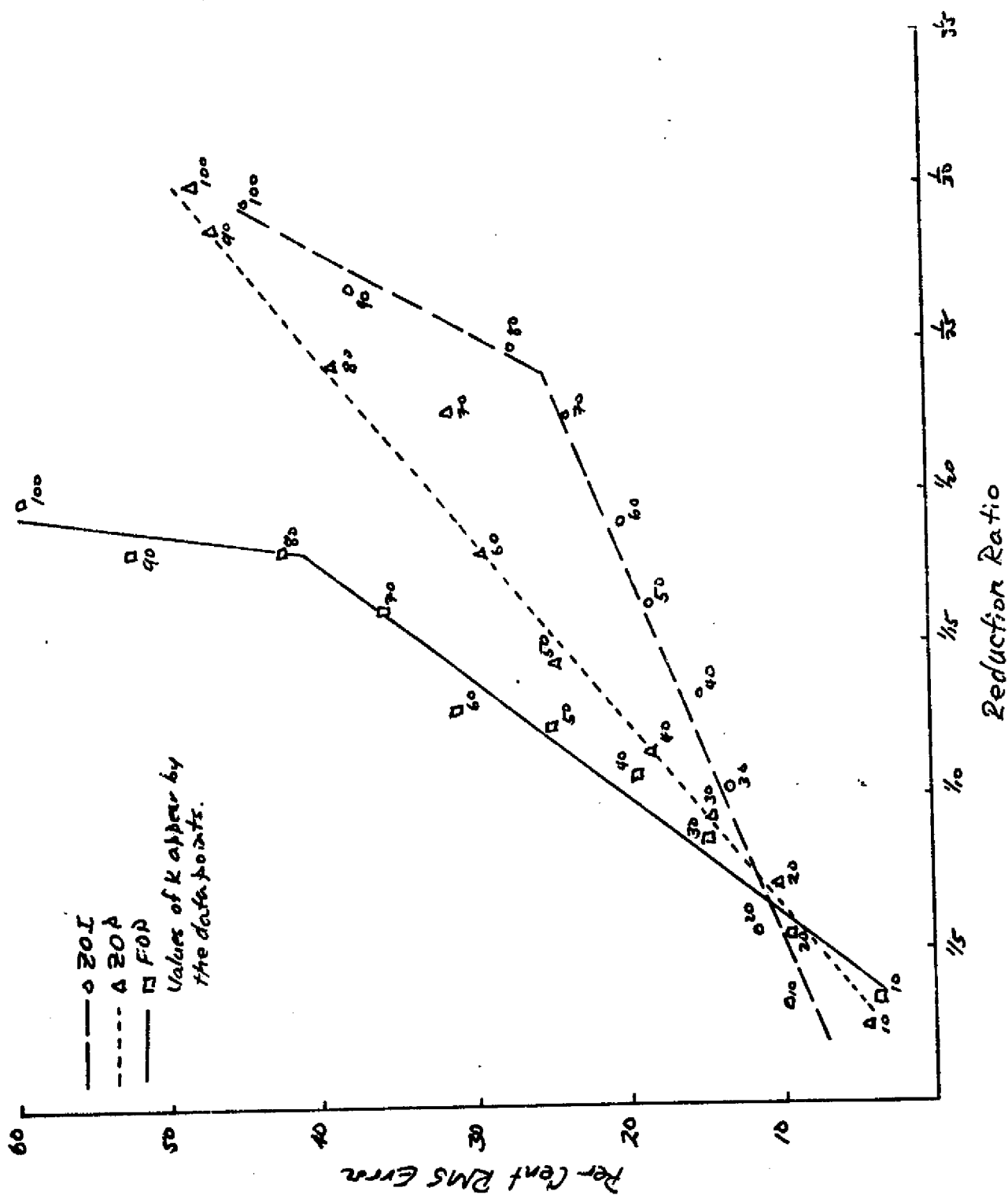
and the RMS Error is given by,

$$\sqrt{\frac{\sum_{i=1}^N (V_i^S - V_i^R)^2}{N}}$$

Given the RMS Signal Value and the RMS Error, the Per Cent RMS Error is given by,

$$\frac{\text{RMS Error}}{\text{RMS Signal Value}} .$$

Figure 10 - Plot of Reduction Ratio vs. % RMS Error. The reduction ratio and % RMS error as defined in the text. Each point in the plot was produced by executing a ZOP, FOP, and ZOI algorithm on the same set of input data. The value of K , the aperture parameter was varied from 10 to 100. The reader should recall that for a aperture parameter of K , the ZOP and FOP set aperture widths of $2K$, while the ZOI sets an aperture width of K .



ORIGINAL PAGE IS
OF POOR QUALITY

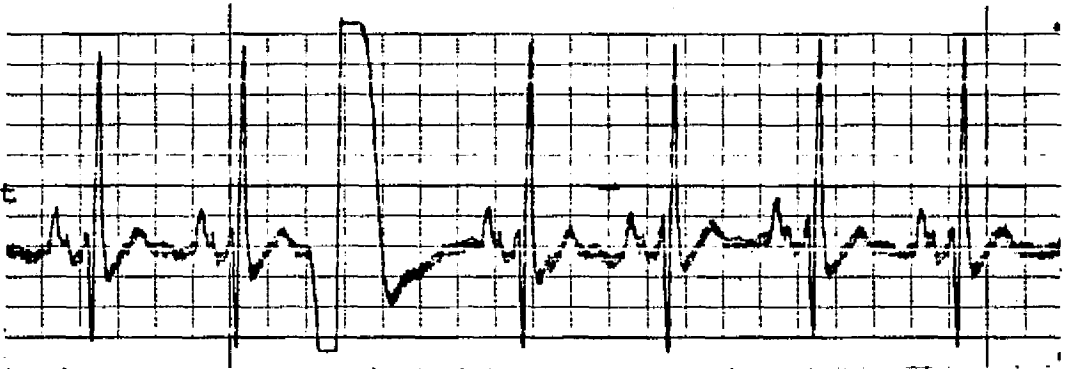
Each data point in Figure 10 was generated by specifying a value for the aperture parameter, K . The reader should recall that the aperture parameter specifies the maximum amount of allowed error before a new approximation is generated. Ten runs were conducted, over the same input data, with K ranging from 10 to 100. The peak to peak signal range was about 600 units.

Figures 11 through 21 illustrate the output of each algorithm for each value of K . The EKG data presented in these stripchart recordings is typical of the data used to produce Figure 10. For each figure, 11 - 21, the stripchart recordings presented are a) the original data, b) ZOI reconstruction, c) ZOP reconstruction, and d) FOP reconstruction.

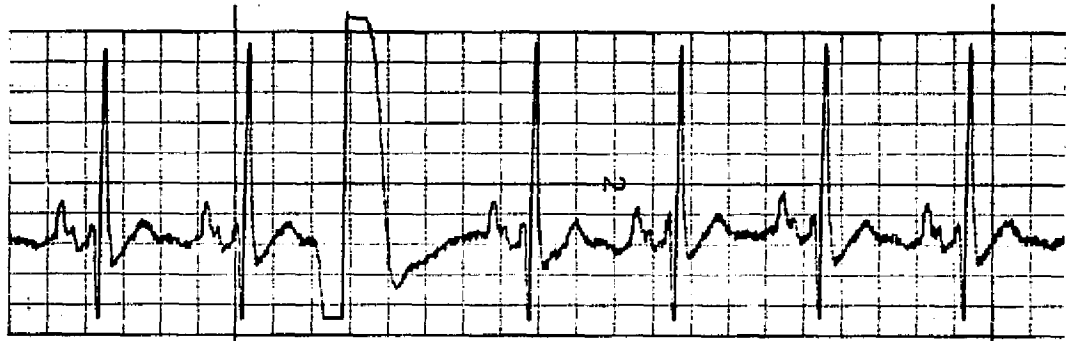
Examination of the results of Figure 10, and the output presented in Figures 11 through 21 indicates that the ZOI is the best of the algorithms. The ZOI is least sensitive to the value of the aperture parameter and produces the most pleasing representation for all values of K .

Figure 11 - Reconstructed ZOI, ZOP, and FOP Output for $K = 10$. This is the first of a series of figures for different values of the aperture parameter K . All the figures contain 4 stripchart recordings; a) the input signal, b) the ZOI output, c) the ZOP output, and d) the FOP output. The narrow spikes in the output of the FOP during the VPB are the result of a digital to analog converter overranging, and are not a result of the FOP algorithm.

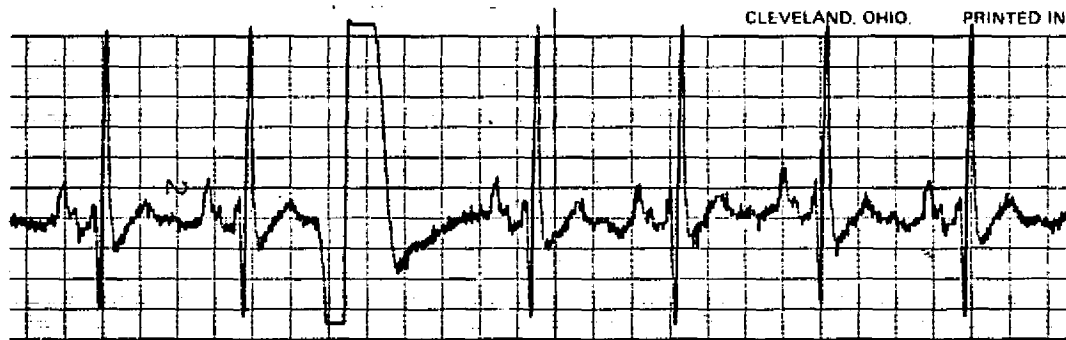
a)
Input



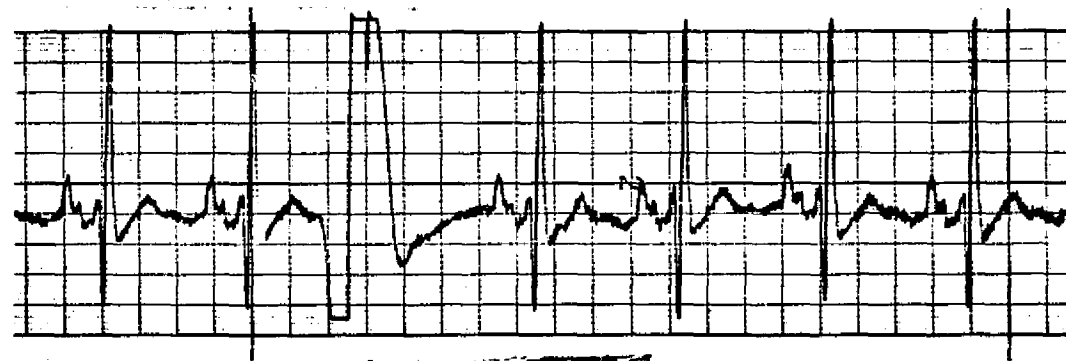
b)
ZOI



c)
ZOP



d)
FOP



CLEVELAND, OHIO. PRINTED IN

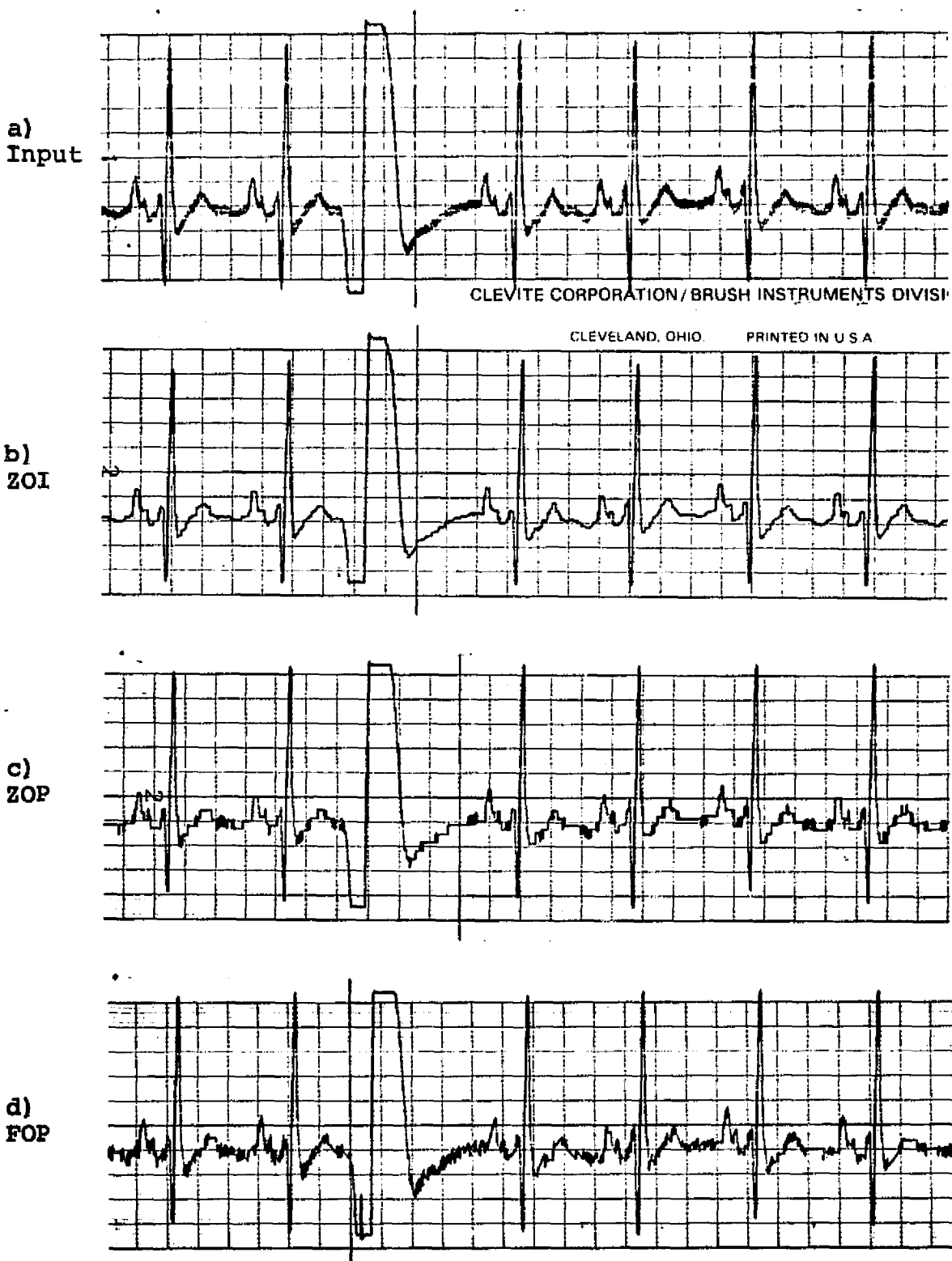
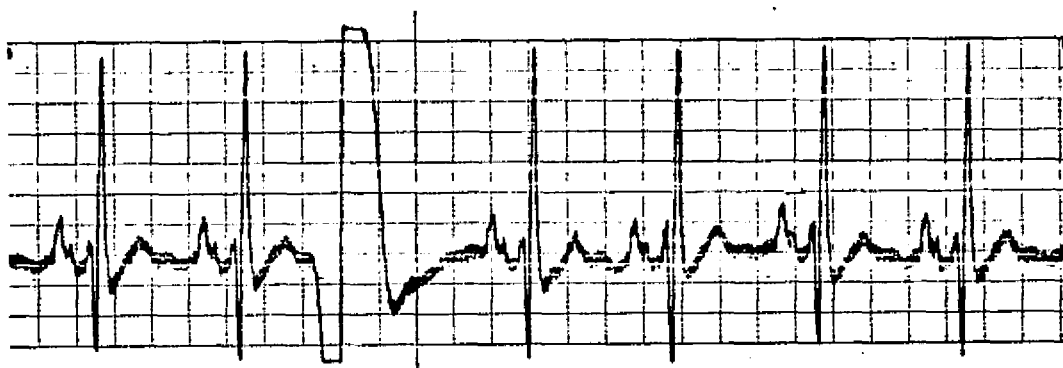
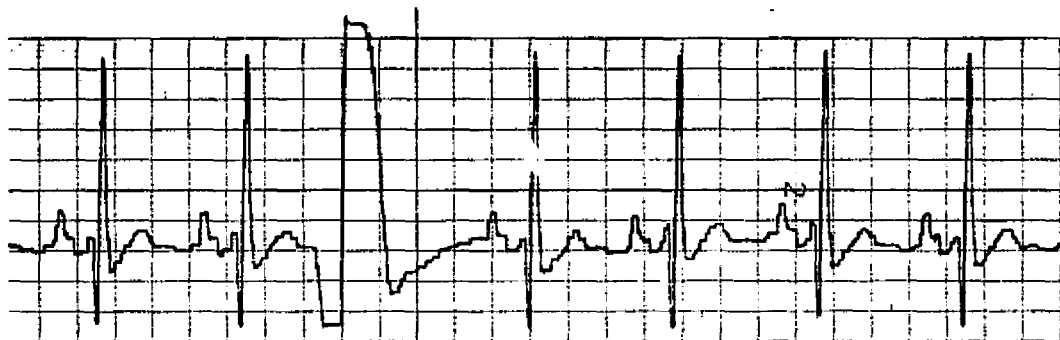


Figure 12 - Reconstructed ZOI, ZOP, and FOP Output for $K = 20$.

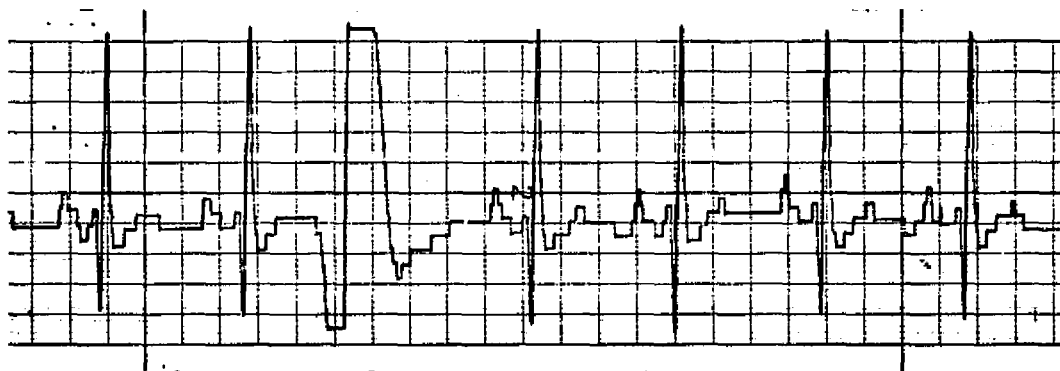
a)
Input



b)
ZOI



c)
ZOP



d)
FOP

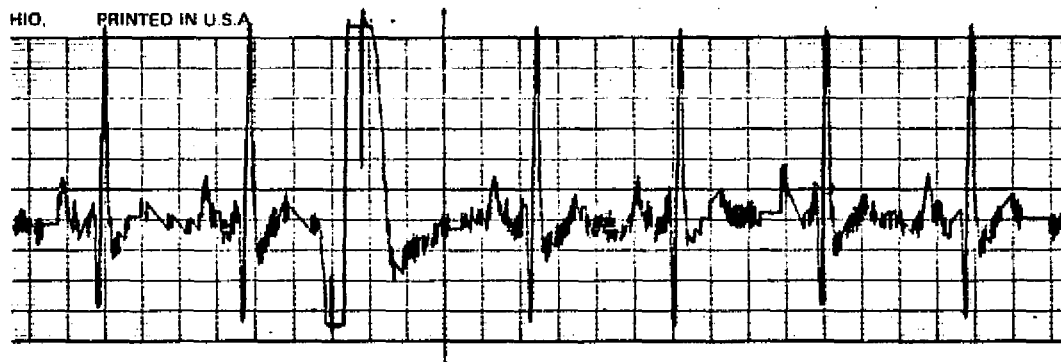
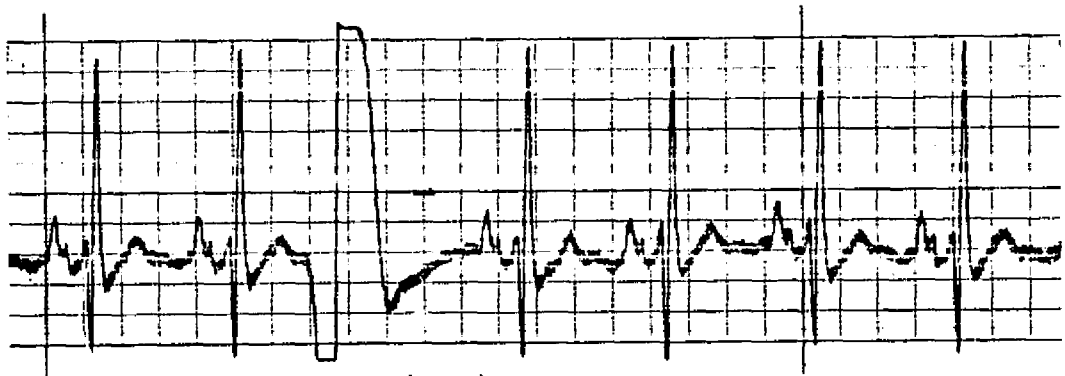
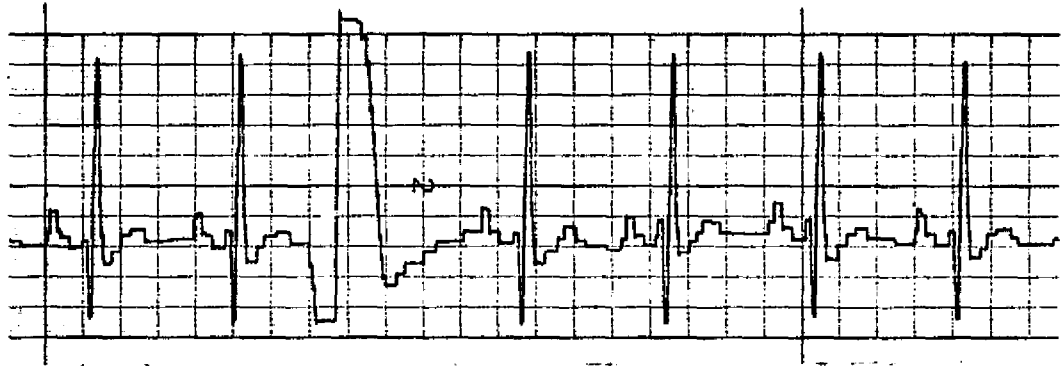


Figure 13 - Reconstructed ZOI, ZOP, and FOP Output for $K = 30$.

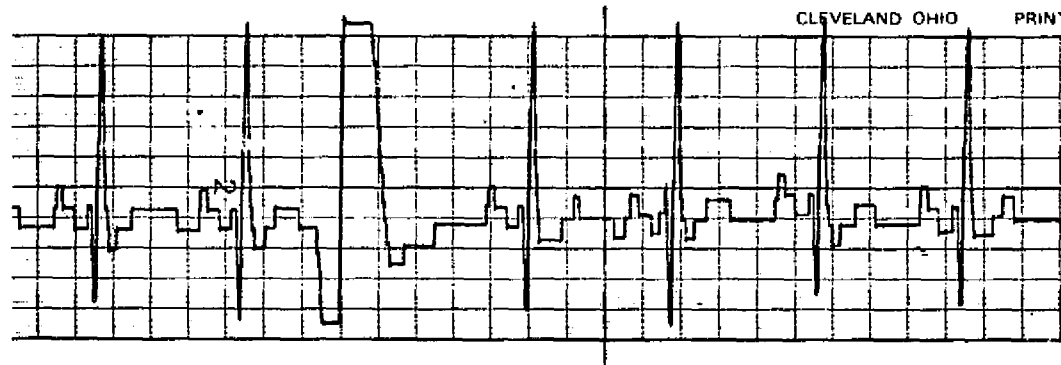
a)
Input



b)
ZOI



c)
ZOP



d)
FOP

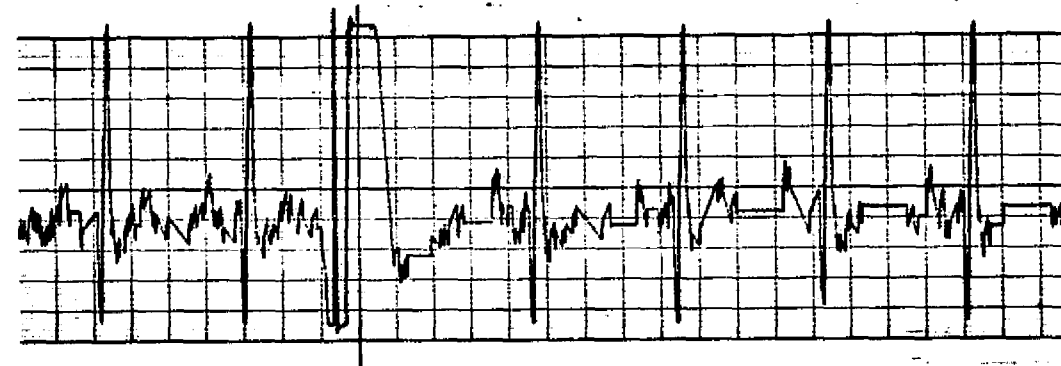
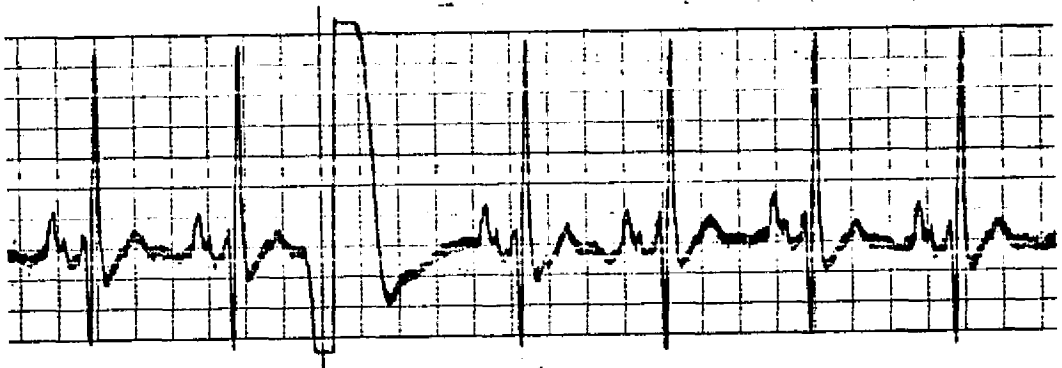
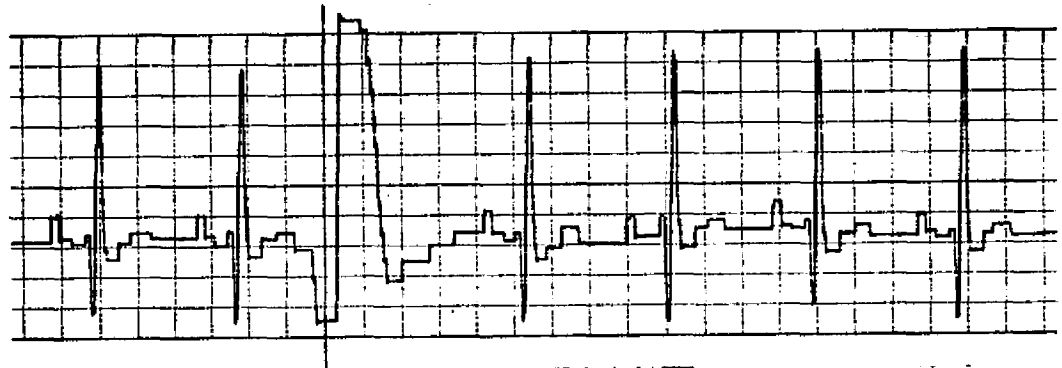


Figure 14 - Reconstructed ZOI, ZOP, and FOP Output for $K = 40$.

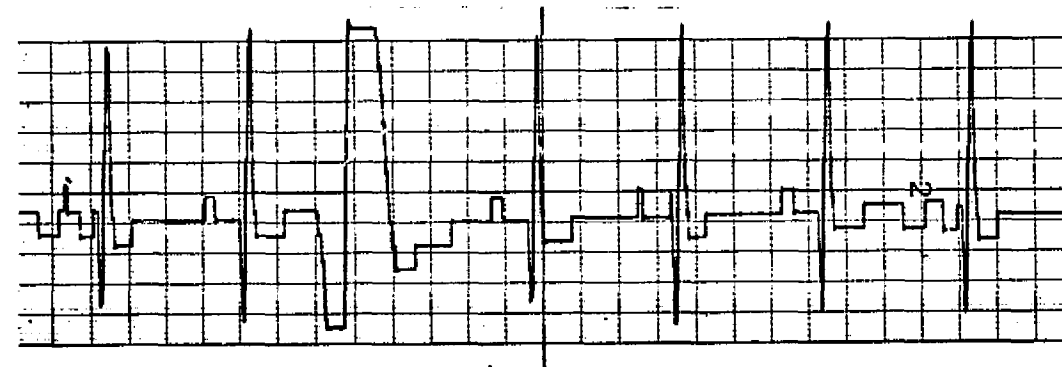
a)
Input



b)
ZOI



c)
ZOP



d)
FOP

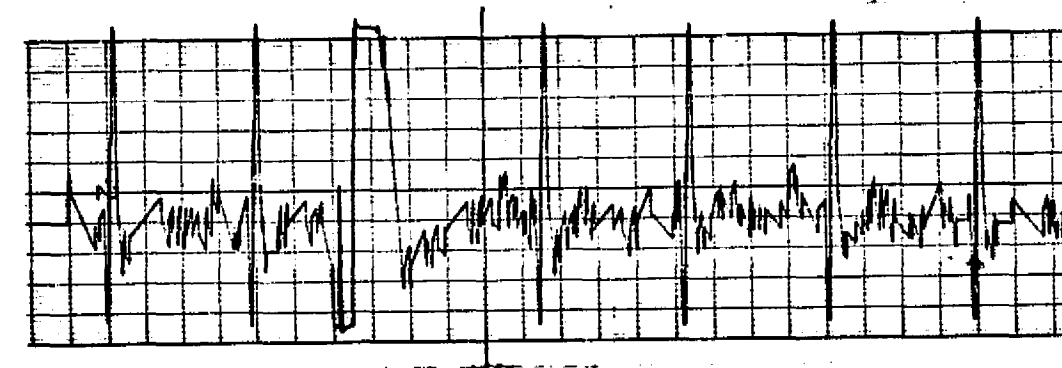
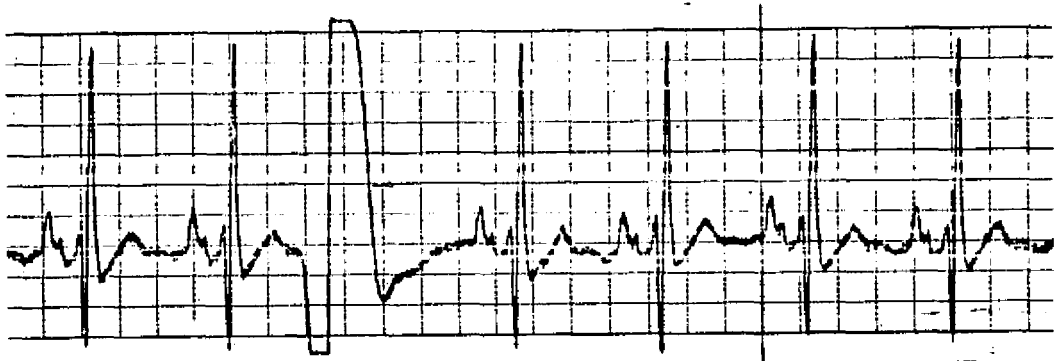
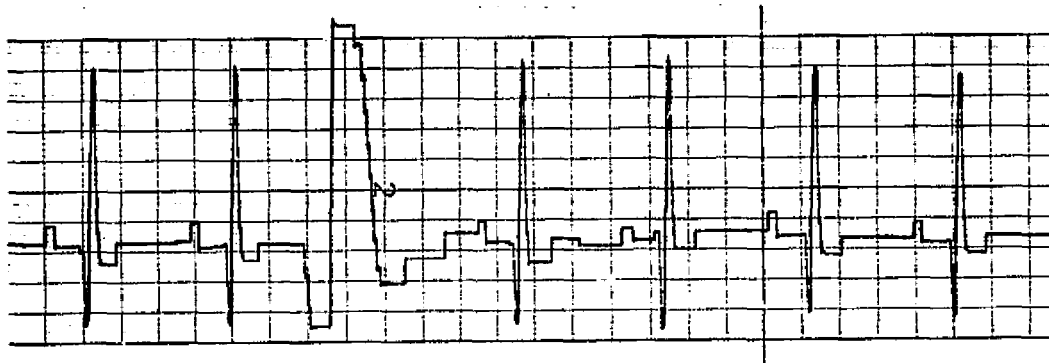


Figure 15 - Reconstructed ZOI, ZOP, and FOP Output for $K = 50$.

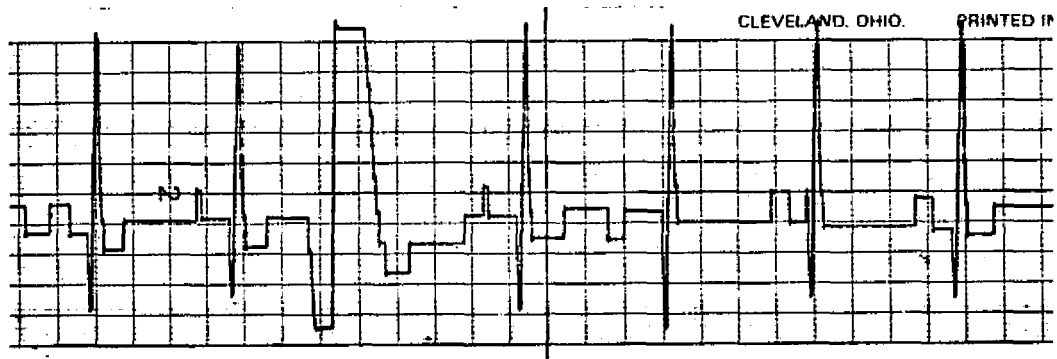
a)
Input



b)
ZOI



c)
ZOP



d)
FOP

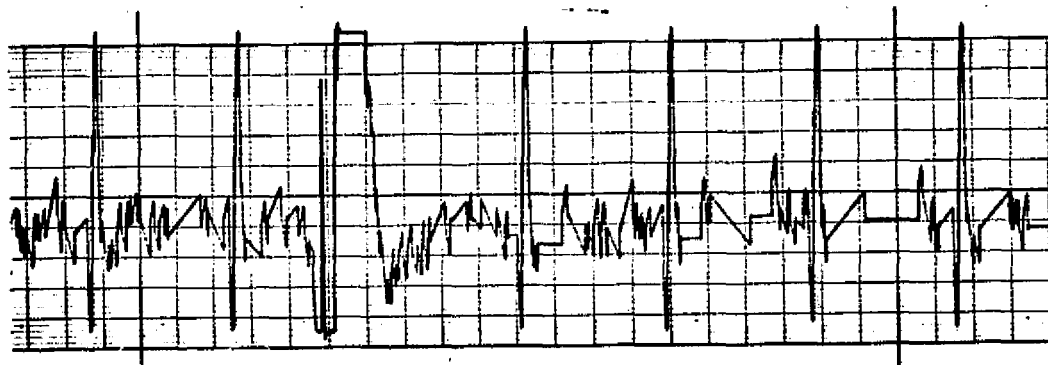
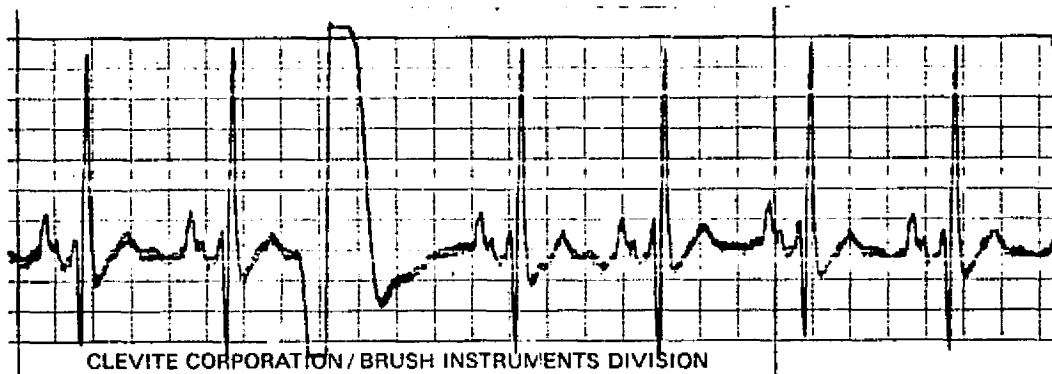


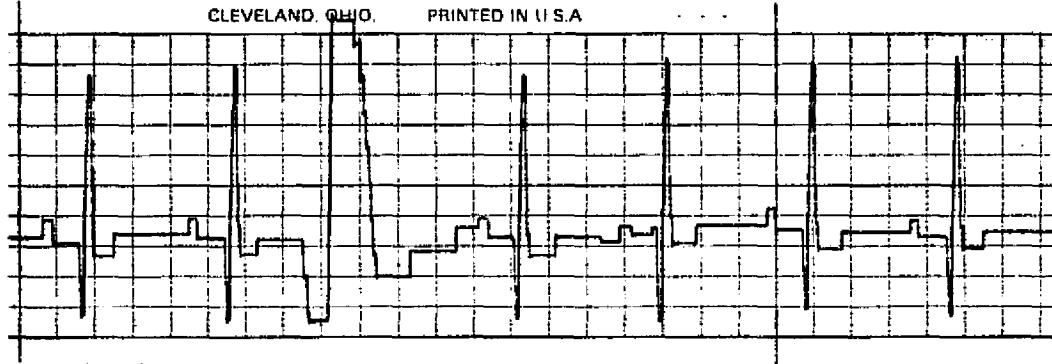
Figure 16 - Reconstructed ZOI, ZOP, and FOP Output for $K = 60$.

a)
Input

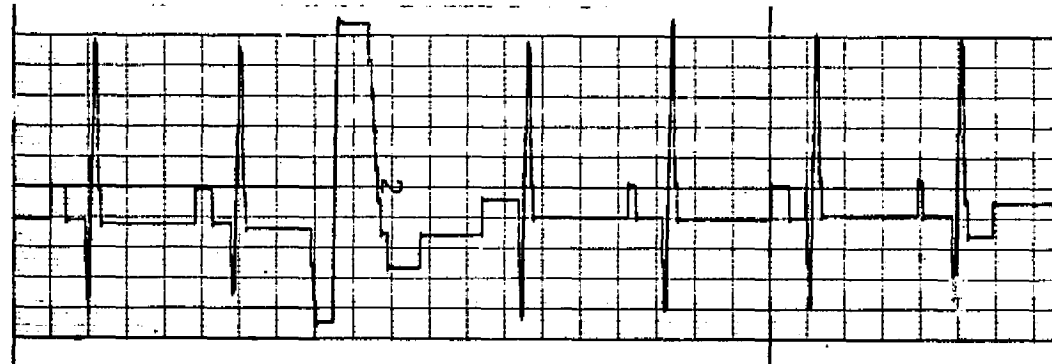


CLEVELAND, OHIO. PRINTED IN U.S.A.

b)
ZOI



c)
ZOP



d)
FOP

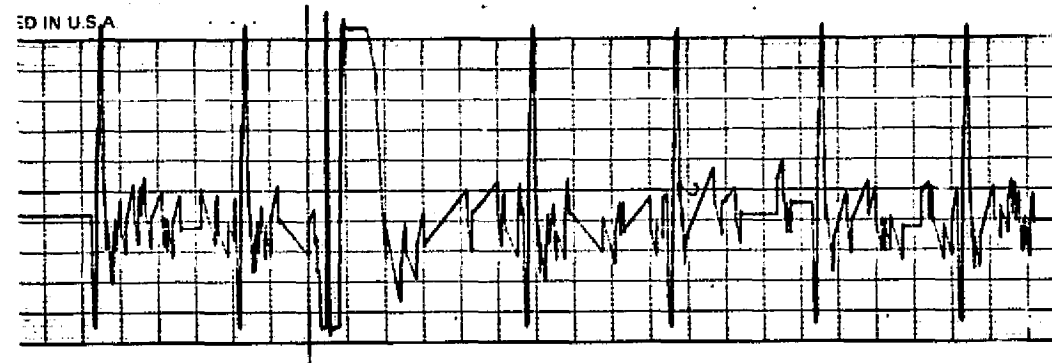


Figure 17 - Reconstructed ZOI, ZOP, and FOP Output for $K = 70$.

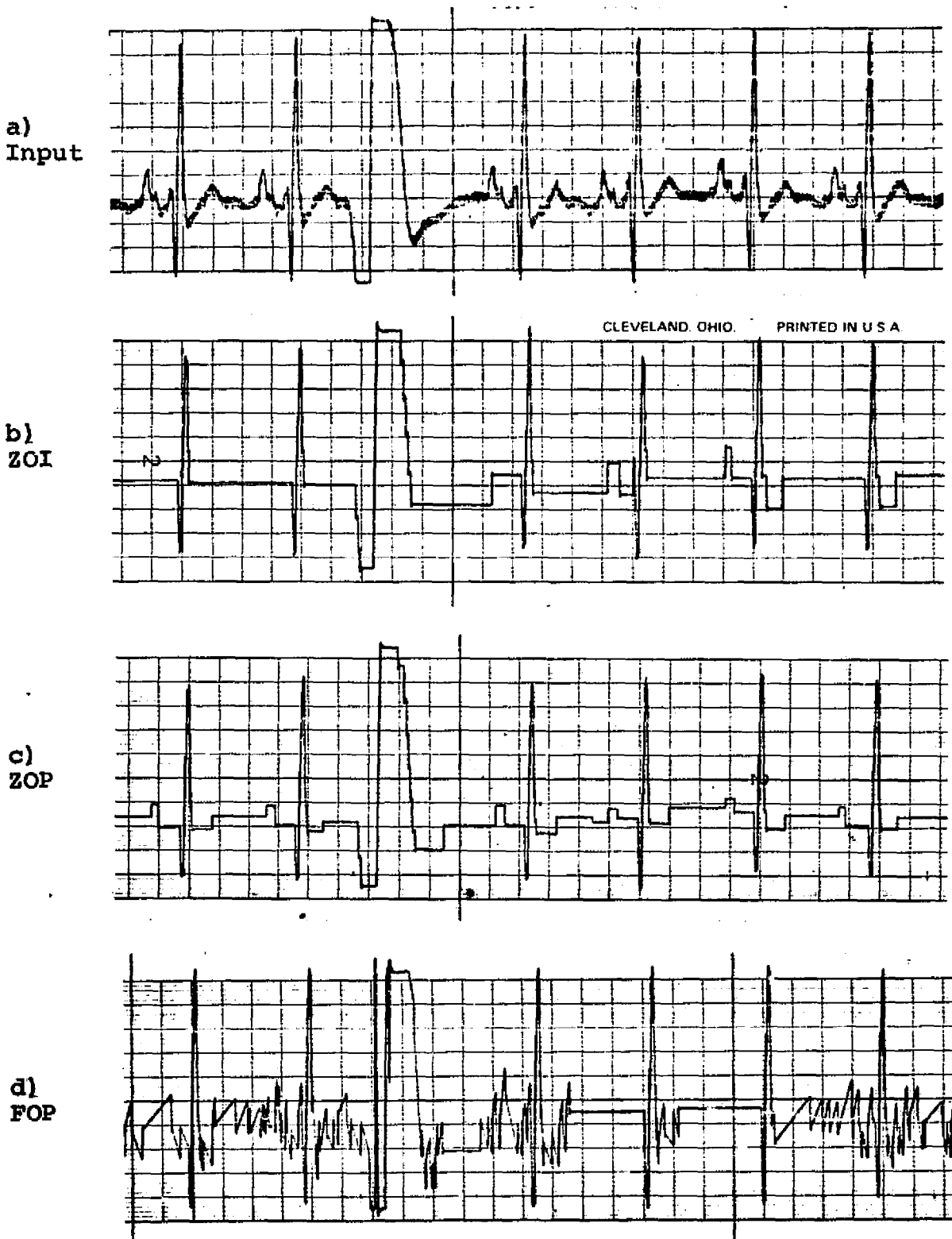


Figure 18 - Reconstructed ZOI, ZOP, and FOP Output for $K = 80$.

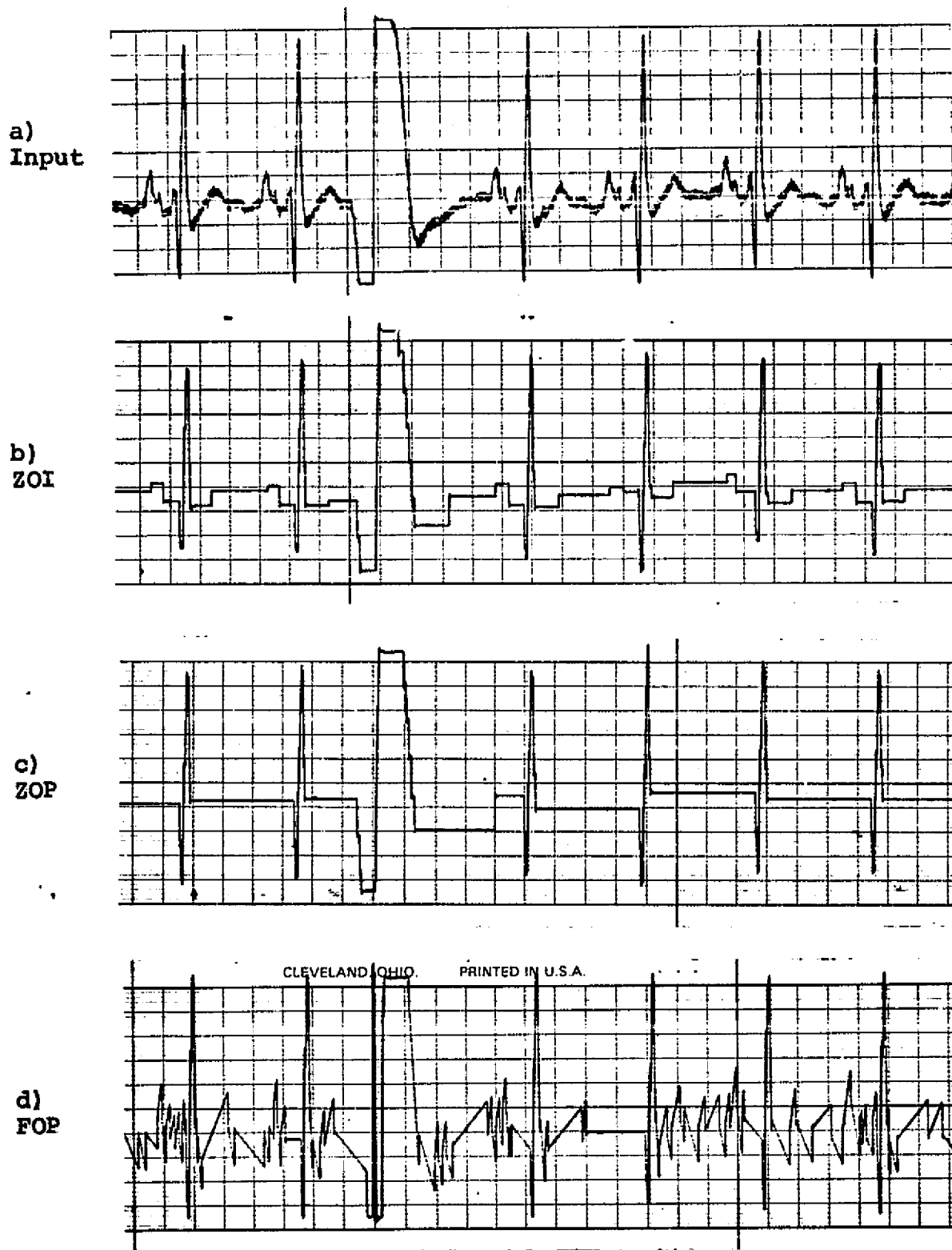
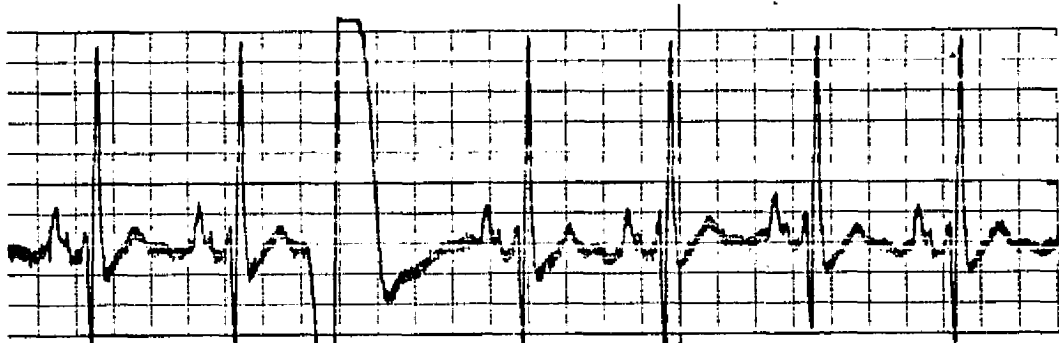


Figure 19 - Reconstructed ZOI, ZOP, and FOP Output for $K = 90$.

a)
Input

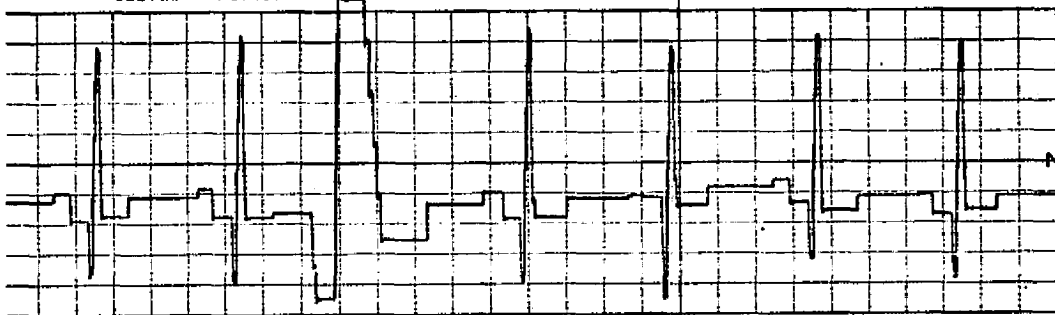


LEVITE CORPORATION / BRUSH INSTRUMENTS DIVISION

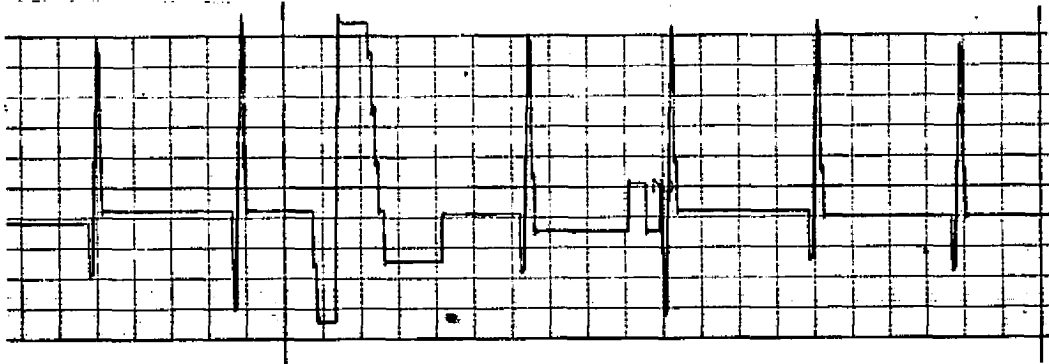
CLEVELAND, OHIO.

PRINTED IN U.S.A.

b)
ZOI



c)
ZOP



d)
FOP

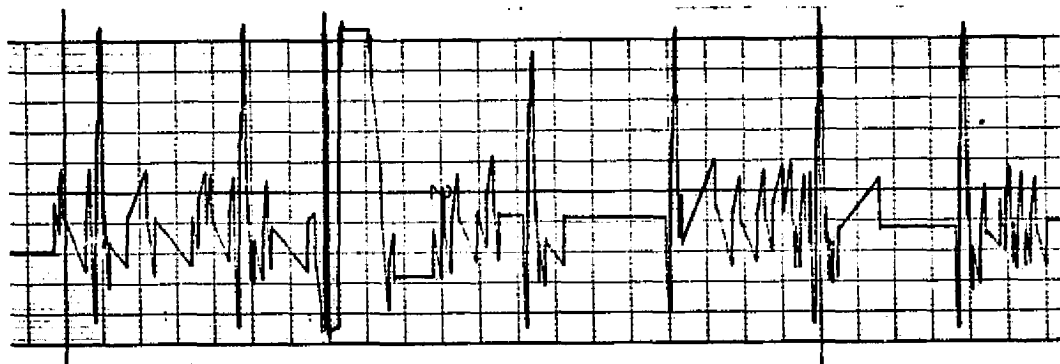


Figure 20 - Reconstructed ZOI, ZOP, and FOP Output for $K = 100$.

CHAPTER III

ALTERNATIVE IMPLEMENTATIONS OF THE
ZERO ORDER INTERPOLATOR

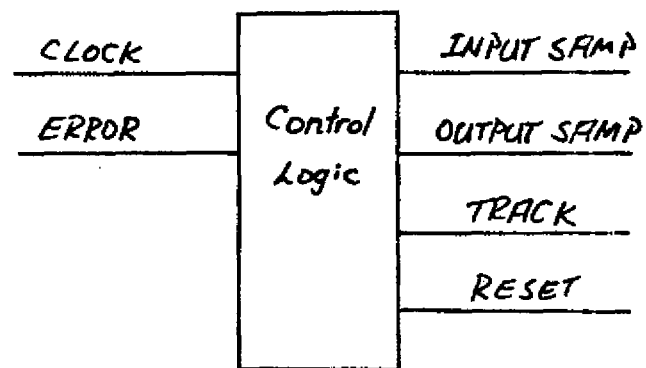
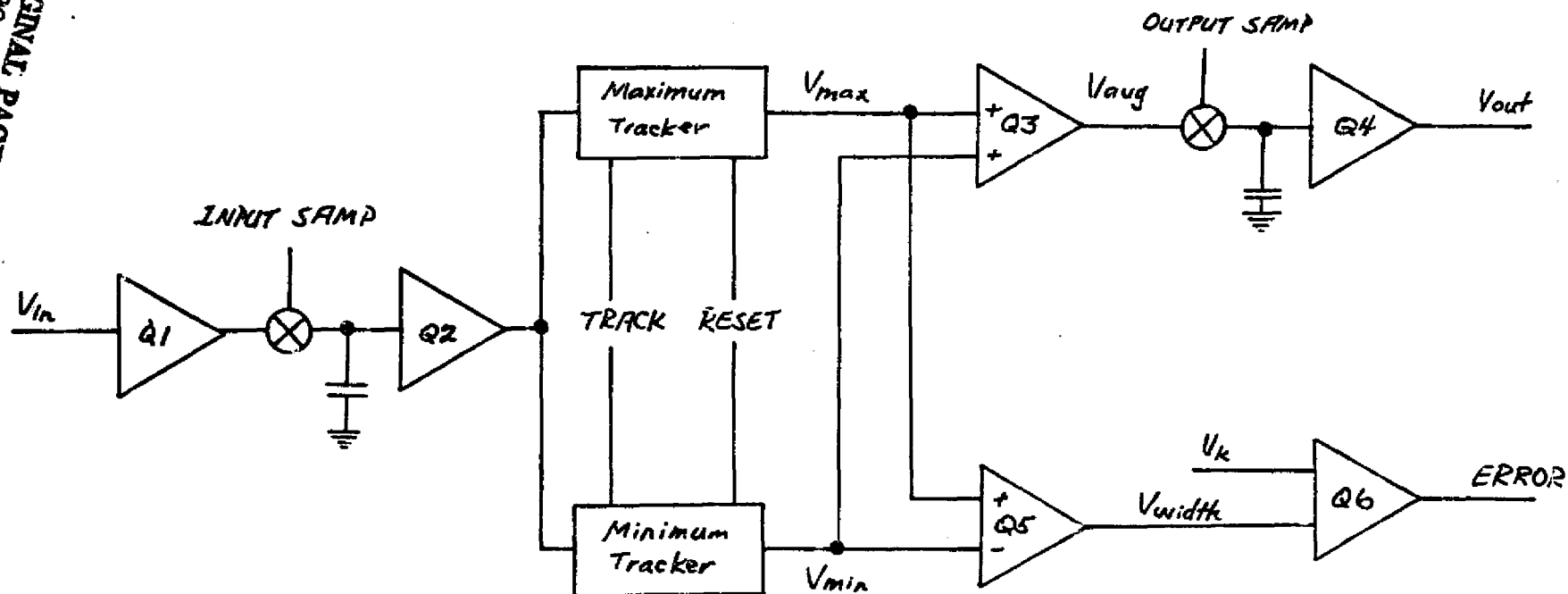
Hardware Implementation:

A hardware system has been constructed which executes the zero order interpolation algorithm. A block diagram of the system appears in Figure 21. A brief discription of the hardware follows, for a detailed discussion the reader is referred to the Appendix.

Referring to Figure 21, amplifier Q1 provides gain and offset to condition the input signal for the following stages. This conditioned signal is sampled by the sample and hold amplifier Q2, which drives the maximum and minimum trackers. These trackers hold their current value until strobed by the TRACK command, at which time they acquire the new maximum or minimum voltage as required. The trackers may also be reset to the current sample voltage by the occurrence of a RESET command. Amplifier Q3 computes the average of V_{\max} and V_{\min} , that is the current interpolated value for the aperture. Amplifier Q5 computes the current aperture width, and amplifier Q6 compares this width against the maximum allowed width, V_k . Thus a control level ERROR is provided when the aperture is too wide. Finally the Control Logic synchronizes the activity of the system with an external CLOCK and provides the required internal timing and gating functions.

Figure 21 - Block Diagram of Hardware ZOI Implementation. This figure presents a block diagram of the hardware ZOI implemented by the author. This block diagram is functionally equivalent to the actual hardware; however, the reader is referred to the Appendix for a block diagram of the true system.

ORIGINAL PAGE IS
OF POOR QUALITY

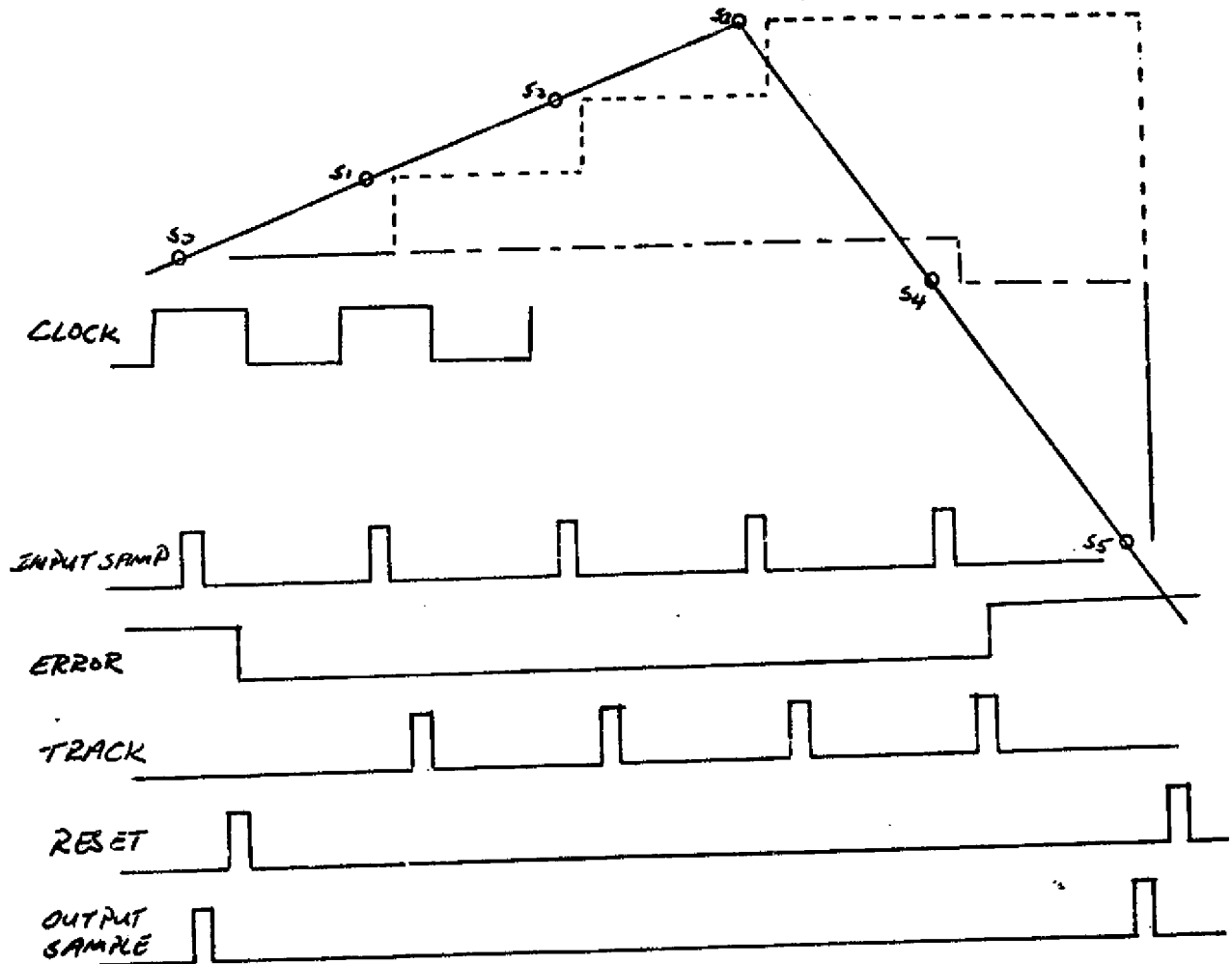


JBW

Figure 22 indicates the sequence of events in the system for a typical input signal. Assume that the sample point s_0 causes the system to terminate the previous segment and begin processing a new series of samples. Since RESET was high after S_0 , V_{\max} and V_{\min} have taken on the value of S_0 . The leading edge of CLOCK initiates the next cycle starting with the acquisition of a new sample value, S_1 . Since the signal has increased in amplitude since the last sample, V_{\max} takes on the new value at the occurrence of the TRACK command. V_{\max} continues to increase for samples S_2 and S_3 . When S_4 is taken in, the value of V_{\min} is decreased. This change produces an aperture width which is too wide, and the ERROR level is raised. The occurrence of an ERROR signal forces the sampling of V_{avg} , thus generating the interpolated output value for the previous series of input samples. A RESET command is given after the next sample value, S_5 , is taken in. This resets the aperture width to zero.

Figure 23 presents output of the system in the form of a stripchart recording of the direct output of Q4. For the waveform presented, a triangle wave, the reconstructed signal and the output of Q4 are similar, however the reader is cautioned that they are not the same thing. A memory, to buffer segment amplitudes and lengths (which the hardware described does not include), is required to generate the reconstructed signal.

Figure 22 - Basic Hardware Timing Diagram. The reader is referred to the text for a discussion.

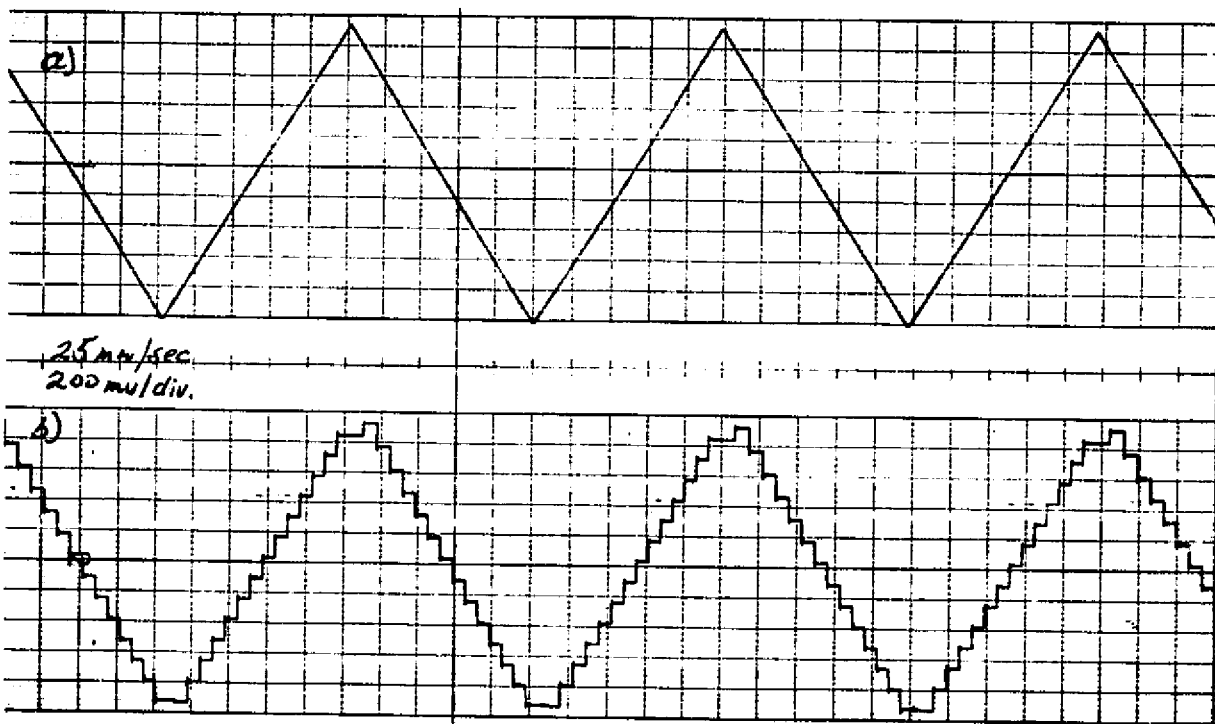


Notes:

- Sampled Data Points
- V_{max}
- V_{min}

ORIGINAL PAGE
OF POOR QUALITY

Figure 23 - Example of Direct Output of the Hardware.
A triangle wave of 0.5 Hz. is the input
signal (a), the output is presented in (b).
The reader is reminded that this is the
direct output not the reconstructed output.



Processor Overhead for Alternative Implementations of the Zero Order Interpolator:

In the context of processing the output of a ZOI on a small computer, the percentage of central processor unit (CPU) time that must be devoted to execution of the ZOI algorithm is important. One would like to make as much time as is possible available to analysis of the data, rather than just compressing input data.

We will consider the percentage of CPU time required for three possible implementations of the ZOI algorithm. The following three implementations will be considered;

- I. Direct software controlled sampling of each signal channel with a software implemented ZOI processor.
- II. Specialized hardware controlled sampling of each signal channel with memory access through a direct memory port and a software implemented ZOI processor.
- III. Specialized hardware implementation of a ZOI processor with software controlled transfers of segment amplitudes.

In addition, we will consider the cost of conventional sampling as a reference point.

Table 1 presents the execution time of various software processes required to implement the above systems. The times are based on assembly language programs written by the author and the timing information provided in (4) for the NOVA 1200.

In addition to the timing information of Table 1, some assumptions about the nature of the source signal must be made in order to estimate the performance of the software and

Table 1 - Software Event Timing

<u>Event</u>	<u>Time to process event (μsec)</u>
a) Interrupt from any external device	40
b) Input sample changes Vmax	37
c) Input sample changes Vmin	40
d) Input sample changes Vmax and puts aperture out of bounds	49
e) Input sample changes Vmin and puts aperture out of bounds	51
f) Buffering output points of ZOI	30
g) Identification of channel number in hardware ZOI	11
h) One data transfer by direct memory port	1.2

All times are based on assembly language programs written by the author. Times are execution time on a NOVA 1200 (4).

and hardware. These assumptions are;

- I) A randomly selected sample point is equally likely to change V_{\max} as V_{\min} .
- II) Every input sample will change either V_{\max} or V_{\min} .

Assumption I is straightforward. It should be noted that assumption I implies that a randomly selected ZOI output segment is equally likely to terminate on a change in V_{\max} as V_{\min} . Assumption II is, in fact, not true; but is made to simplify the estimation of expected processing time per input point. It can be justified on the grounds that the software requires about as much time to process a point that does not change V_{\max} or V_{\min} as it does to process a point that does change one of them.

A general purpose computer with analog to digital conversion equipment and an analog channel multiplexer forms the hardware of implementation I. The software costs can be broken down into three areas, (1) interrupt overhead to divert the processor from any other task it might be performing, (2) the expected processing time in the ZOI software for each sample point, and (3) the expected buffering time to save any output from the ZOI software. The time required to respond to an interrupt is fixed at 40 μ sec. The actual time required in the ZOI software is a random variable, however using the assumptions presented earlier, and specifying a reduction ratio, R (see the previous chapter), we may compute the expected value of the processor time required per input sample to be,

$$40 + (38.5 + 41.5R)N \text{ microseconds,}$$

where N is the number of active channels that must be compressed. This expected value may be converted in to percentage of CPU time by dividing by the period between input samples.

System II uses some what more complex hardware in an effort to avoid the relatively high interrupt overhead (for small numbers of channels) by placing the sampled data in memory through a direct memory port. This avoid the interrupt overhead for each point sampled, since an interrupt would not be generated until the core buffer was full. One disadvantage of this method is that it does require buffering the original data and thus requires core that Systems I and III would not need. Referring to the timing information of Table 1 and the assumptions of this section, the expected value of the processor time required to handle each sample point is,

$$(39.7 + 46.5R)N \text{ microseconds, where}$$

where R is the reduction ratio, and N is the number of active channels. As can be seen, this method offers little improvement over the previous system.

A specialized hardware implementation, as described in the previous section, comprises system III. In this case the processor time per output of the ZOI hardware is fixed at 81 microseconds (interrupt overhead, buffering, and channel identification) per output point. Since the fraction of sample points expected to result in output of a segment is given by R, the expected value of processor time per input sample

point is,

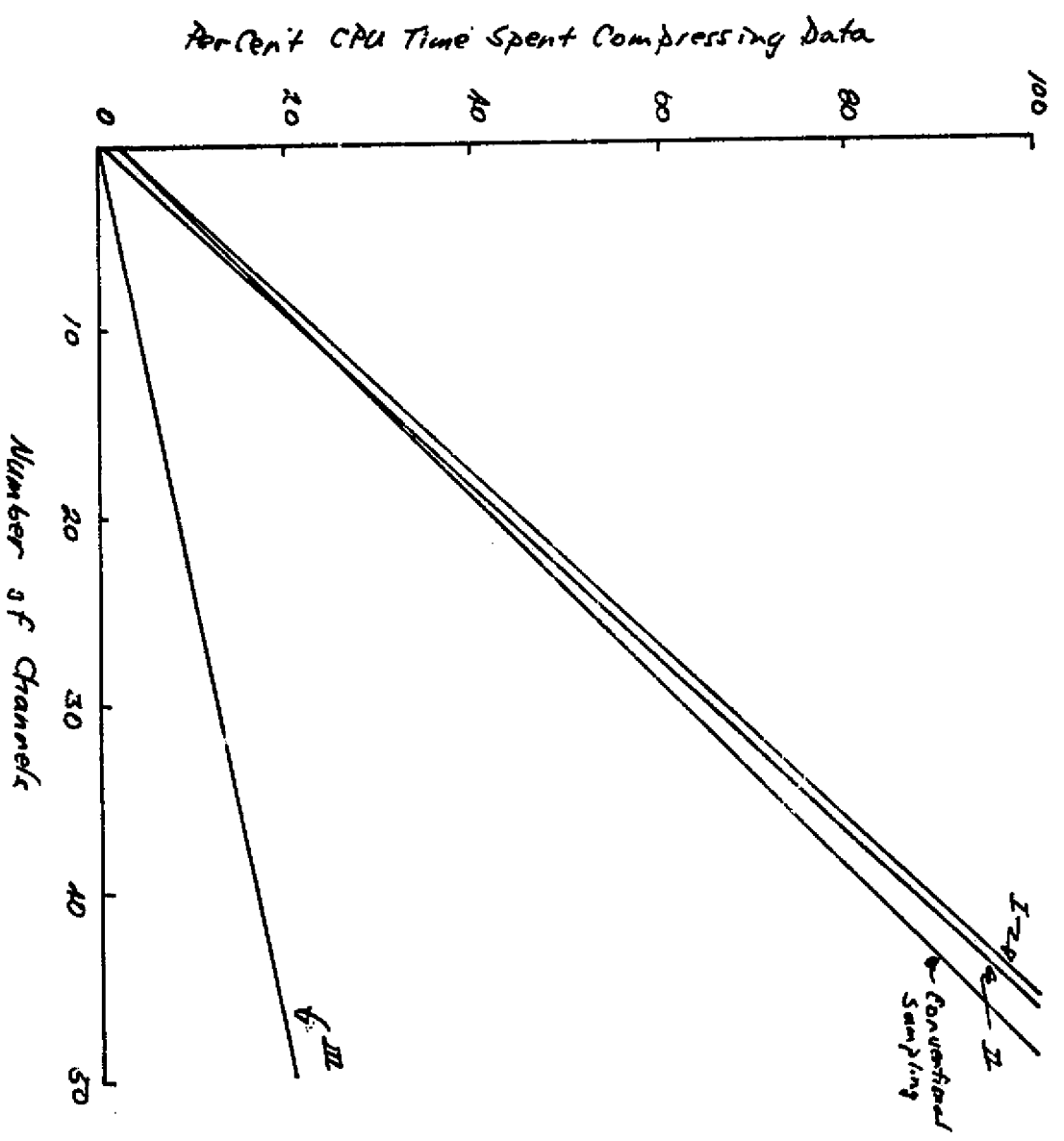
$81RN$ microseconds,

where R is the redundancy reduction ratio, and N is the number of active channels.

The alternative systems are compared in Figure 25 for $R = 0.1$ and a basic sample interval of 2000 microseconds (500 Hz.). The plot presents percentage CPU time vs. the number of active channels. In addition to the three systems discussed in detail, the overhead for conventional sampling is presented.

Both implementations involving software ZOI processing are comparable to conventional sampling, while the hardware implementation of the system is superior.

Figure 24 - Per Centage of CPU time vs. Number of active Channels. The plot is for an assumed reduction ratio of 0.1 and a basic sampling rate of 500 samples per second. The per cent axis may be scaled for other sampling rates. For instance for a sampling rate of 250 samples per second, the value of 20% becomes 10%.



ORIGINAL PAGE IS
OF POOR QUALITY

C.2

CHAPTER IV

CONCLUSIONS

General:

The object of this work was to determine an efficient and practical representation of the EKG for digital processing.

The results presented in Chapters II and III indicate that the zero order interpolator provides an efficient representation of the EKG and is ^apracticable to implement, both in software and hardware.

Examination of the records presented in Figures 11 through 21 indicates that for moderate values of K (around 30) the ZOI provides an excellent representation of the original signal. Certainly no cardiologist would have any difficulty making a rhythm analysis from the results presented. Redundancy reduction ratios of 0.1 to 0.05 ~~for~~ moderate values of K indicates that the representation is indeed more efficient than conventional sampling.

The results of Chapter III, in particular the data summarized in Figure 25, indicates that the implementation of the ZOI algorithm in software places a burden on a processor roughly equivalent to the demands of conventional sampling. If the algorithm is executed in special hardware the processor demands are negligible for any reasonable number of channels.

APPENDIX

HARDWARE DISCUSSION

General Information:

The hardware design for the zero order interpolation algorithm uses both analog and digital integrated circuits. The analog circuitry is constructed on two Vector boards (Vector Electronics Co., Inc.), while the digital logic is constructed on CBG-5 logic boards. The CBG-5 boards have space for three 14 or 16 pin dual in-line packages. The edge connection is 44 readout, and all package pins other than power and ground are available at the edge connector. The cards were assembled in a card rack, and all interconnections are via AMP Taper Pins (AMP Inc.). The analog functions were implemented with operational amplifiers; the LM301A differential amplifier, the LM310 follower, and the LM311 comparator (National Semiconductor Corp.). These amplifiers were chosen for their high performance/cost ratio. The control logic was implemented with series 74 transistor-transistor logic (Texas Instruments, Inc.). This logic was chosen because of its availability and the medium scale integration functions, such as counters and decoders, included in the series.

Figure A1 provides a functional schematic of the system. The hardware divides into two major sections, the analog circuits and the control logic. The analog circuits provide functions such as sample and holding, maximum tracking, minimum tracking, and voltage comparison. The control logic accepts inputs from an external CLOCK and the analog circuitry

and generates commands to the analog circuitry.

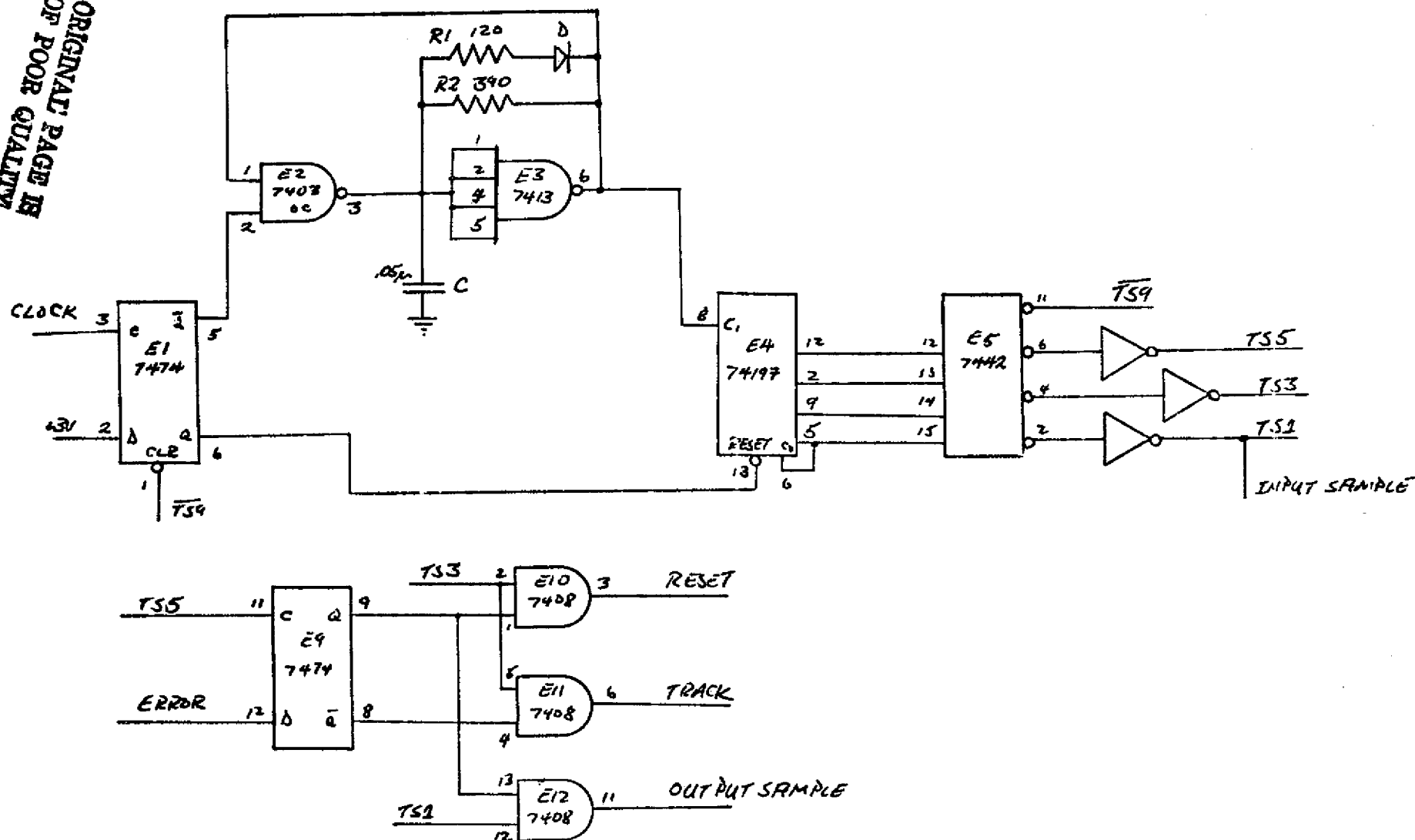
System Control Logic:

The control logic is presented in Figure A2 and an associated timing diagram is presented in Figure A3. The control logic generates a series of time states (TS0, TS1, ..., TS9) which are gated with other logic levels to provide commands to the analog circuitry.

Assume the system is waiting for the next CLOCK signal to start a new cycle. Referring to Figure A2, consider element E1 (a D type flip-flop) to be in the low state ($Q=0$). Its Q output thus asserts the reset function of element E4 (a 4 bit counter). Since counter E4 is reset, all its outputs are low and element E5 (a decoder) decodes this as a zero. The ten decoded outputs of E5 correspond to the time states of the system, the the system is now in TS0 (no output is shown on E5 for TS0 since this logic level is not actually used by the system). When a zero to one transition occurs in the CLOCK level, E1 changes to high state. This enables the counter, E4.

Elements E2 (an open collector nand gate) and E3 (a Schmitt trigger) form a gated multivibrator which clocks the system. Ignoring E2 for the moment, we will explain the operation of E3 and its associated passive components. When power is first turned on, C is discharged, thus the input of E3 is low, forcing the output high ($\sim 4.5v$). When the output goes high, C begins to charge up, to 4.5v, through resistor R2. Note that diode d is backbiased so that R1 does not affect the charge up cycle

ORIGINAL PAGE IS
OF POOR QUALITY



18

Figure A2-System Control Logic.

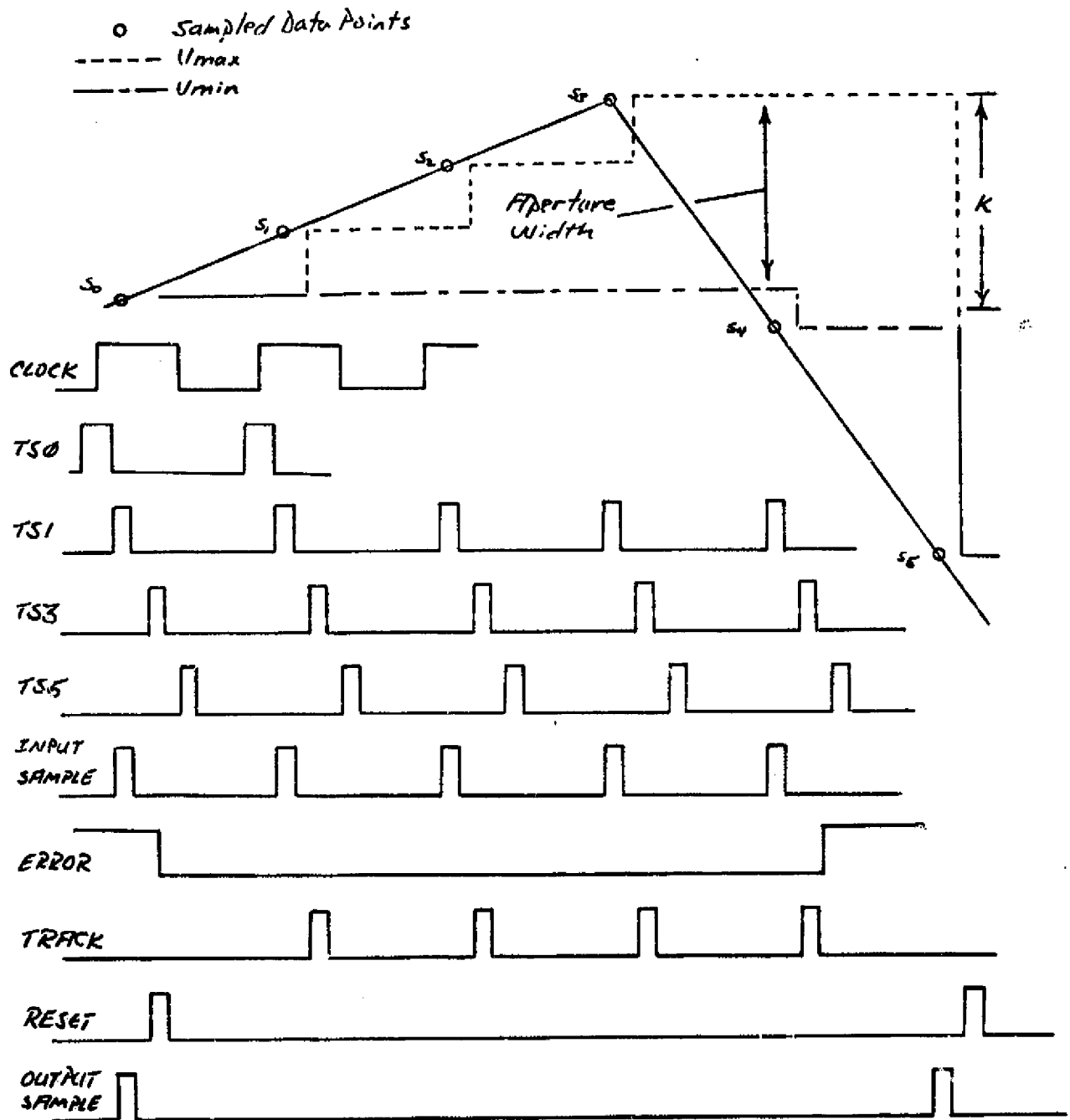


Figure A3 - System Timing Diagram

of C. When C reaches the upper threshold of the schmitt trigger, the output of the trigger will go to logic 0 ($\sim 0.4v$). Now C begins to discharge through both R1 and R2 until it reaches the lower threshold of the schmitt trigger. When this happens, the output will go high, and the cycle will then repeat. Element E2 gates the mutivibrator by grounding the input of E3 when both of its inputs are high. Since input E2-1 is from the output of E3, the multivibrator can only be shutdown when E3 is in the high state. This forces an integral number of pulses of constant width to be produced, with out regard to the relative time the gating input, E2-2, is raised. The values of R1, R2 and c where chosen to provide a clock period of about 20 microseconds, thus each timestate (except TS0) is 20 microseconds in duration.

The clock pulses produced by E3 are counted by E4, and E5 decodes the appropriate time state. When TS9 is decoded, E1 is cleared through the asynchronous clear, pin 1. This shuts off the multivibrator and clears the counter causing TS0 to be decoded. The control control logic will now wait for another external CLOCK to initiate the next cycle.

Time states T1, TS3 and TS5 are gated to produce commands to the analog circuits. Refering to Figure A3, note that ERROR is high at the start of cycle 0. Thus the system error, or aperture width, went out of bound during the previous cycle. During TS5 of the previous cycle, E9 was clocked (Figure A2) into the high state since ERROR was high.

Cycle 0 starts with the CLOCK transition, and TS1 is entered. TS1 always generates an INPUT SAMPLE command since the input signal is sampled every clock cycle. E9 is in the high state from the previous cycle, enabling E12. This results in an OUTPUT SAMPLE command during TS1. OUTPUT SAMPLE forces the output sample and hold to acquire the current value of V_{avg} as the output of the system, V_{out} . V_{out} (Figure A1) now represents the interpolated value of the input signal since the last OUTPUT SAMPLE command. The interval between OUTPUT SAMPLE commands is the length of the segment.

TS3 is entered next. Since E9 is in the high state, E10 generates a RESET command. RESET forces the Maximum and Minimum trackers to acquire the current value of the input sample, S_0 in this case. V_{max} and V_{min} are at the same value so ERROR returns to the low state. TS5 will now clock E9 into the low state.

During cycle 1, The system acquires a new input sample, S_1 , during TS1. No OUTPUT SAMPLE command is generated since E9 is low; however, TS3 generates a TRACK command through E11. For this cycle, V_{max} increases in response to the TRACK command. ERROR remains low, and E9 is clocked low by TS5. This sequence of events repeats for cycles 2 and 3.

V_{min} is changed during cycle 4, and the aperture width increases beyond the limit, V_k , forcing ERROR high. TS5 will clock E9 high, thus forcing an OUTPUT SAMPLE command in cycle 5.

Analog Design:

Signal conditioning for the system is provided by amplifier Q1 of Figure A1. Figure A4 is the schematic diagram of this stage. The stage is of conventional negative feedback design, providing a gain range of 0.1 to 2. An adjustable offset voltage is applied to the positive input of the amplifier through the resistive divider of R5 and R6, providing a DC offset of ± 5 volts. C1 provides feedforward compensation for the amplifier, as recommended by the manufacturer. This increases the slew speed of the amplifier to $10 \text{ V}/\mu\text{sec.}$, resulting in improved bandwidth and step response. A wideband amplifier is not required in this stage and a less expensive amplifier could be substituted.

The analog sample and hold function was required to sample the input voltage (amplifier Q2 of Figure A1) and to hold the output of the system (amplifier Q4). In addition the Maximum and Minimum Trackers use this function. In all cases the sample and hold function was implemented with the circuit of Figure A5. On command the storage capacitor, C1, is charged to the value of the input voltage, V_{in} , through the FET switch, Q1. The LM310 voltage follower allows readout of the stored voltage. Pertinent specifications of the LM310 follower and LS4391 FET switch are:

LM310 Voltage follower--

Input Bias Current	10 nA (maximum)
Input Resistance	10^{10} ohms (minimum)
Offset Voltage	10 mV (maximum)

ORIGINAL PAGE
OF POOR QUALITY

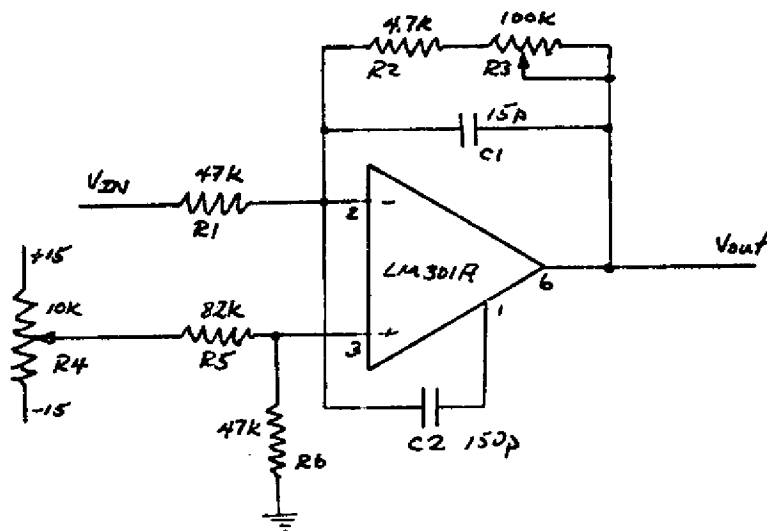


Figure A4 - Signal Conditioning Stage

ORIGINAL PAGE IS
OF POOR QUALITY

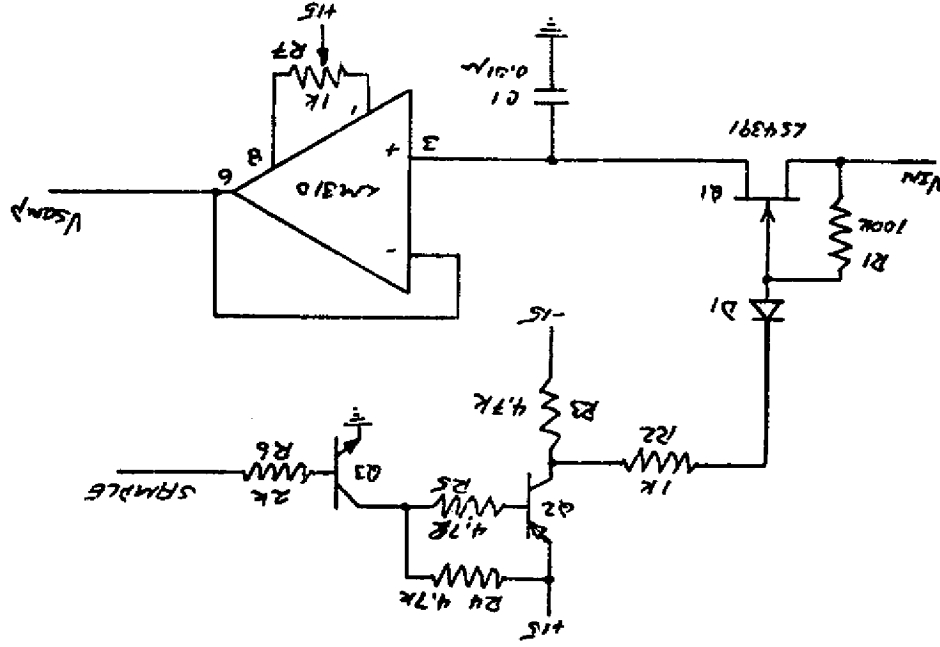


Figure H6 - Sample and Hold

LS4391 (2N4391) FET switch-

R on	30 ohms (maximum)
Off state drain leakage	0.1 nA (maximum)
Pinchoff voltage	10 V (maximum)
Gate-source capacitance	3.5 pf (maximum)

The major considerations in a sample and hold circuit are the acquisition time and droop. Acquisition time is the time required for the sample and hold to come to within a specified error range of the input signal, when switched into track (or sample) mode. Droop is the decay rate of the output voltage while in hold mode.

The FET and capacitor, of Figure A5, form an RC circuit. In the on stage the worst case resistance of the FET is 30 ohms, thus the RC network has a time constant of 0.3 microseconds, with $C_1 = 0.01$ microfarads. This circuit will require about 4.6 time constants to charge to within 1% of the final value, thus the capacitor acquires the input signal to within 1% accuracy in about 13.8 microseconds. The LM310, due to its high slew speed and excellent large signal step response adds little to the overall acquisition time of the sample and hold.

Droop is caused by charge leaking off the storage capacitor. This charge may leak out through the FET switch and the input of the LM310 (provided good capacitors are used the internal leakage of the capacitor is not important). The major source of leakage current is the bias current of the LM310. Given a worst case bias current of 10 nanoamperes, the worst case voltage droop is given by,

$$\frac{dV}{dt} = \frac{I}{C} = \frac{10 \times 10^{-9}}{0.01 \times 10^{-6}} = 1 \text{ volt/second.}$$

Thus in 100 msec., a reasonable maximum holding time, an error of ¹⁰⁰~~20~~ mV may accumulate. For a peak to peak range of 5 volts this would correspond to ²~~1.4~~% error.

The FET is switched by transistors Q2 (Figure A5) and Q3 through diode D1. When SAMPLE is high, Q3 is on and its collector is at ground. This pulls the base of Q2 down through R5, biasing Q2 on. When Q2 is on, its collector is at +15 volts, thus backbiasing^g diode D1. Since^c the drain and source of the FET are connected through R1, they are now at approximately the same voltage and the FET is in its on state.

When SAMPLE is low, Q2 and Q3 are off. The collector of Q2 is pulled to -15 volts through R3. Diode D1 is forward biased for any V_{in} above -15 volts, thus the FET junction is reverse biased. Pinch off voltage for the LS4391 is 10 volts, maximum, so that input voltages above -5 volts will keep the FET in pinchoff.

Considering just the FET, the turn on time is given by the time constant of R1 and the drain-source capacitance (3.5 pf) of the FET, or 0.35 microseconds. The turn off time is determined by the timeconstant of R2+R3 and the drain-source capacitance, or 21 nanoseconds. Both of these times are small compared to the acquisition time and can be ignored in this application.

The maximum tracking function was implemented with the circuit of Figure A6. This maximum tracker is not a continuous

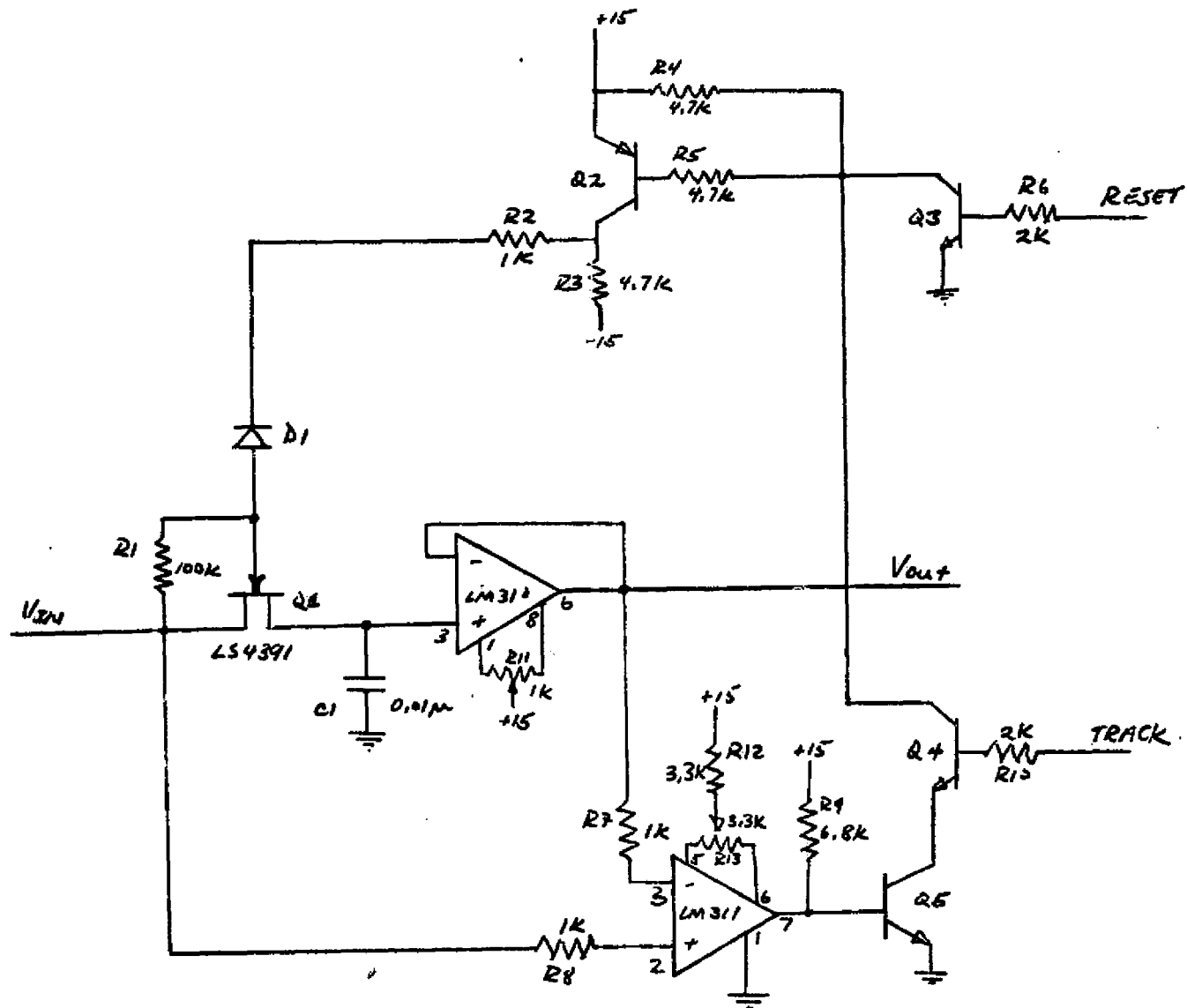


Figure A6 - Maximum Tracker

tracker, that is it can track the changes in the input only when gated by the TRACK command. In addition the tracker may be reset at any time to the current input value through the RESET command.

A major part of the maximum tracker is a sample and hold circuit which was discussed earlier. The circuit for the maximum tracker is presented in Figure A6. The LM311 comparator, compares the input value to the tracker, V_{in} , with the current output, V_{out} . If the input is greater than the output, the output transistor of the LM311 will be off, thus the base of Q5 will be pulled up by R9. Q4 and Q5 form an 'and' gate. If V_{in} is greater than V_{out} (implies Q5 is on) and the TRACK command is high (thus switching Q4 on) then the base of Q2 will be pulled to ground through R5, Q4 and Q5. This will switch Q2 on and thus switch the FET on.

Input offset adjustments (R11, R12, R13) have been used on both the LM310 and LM311. Note that at least one of these offset adjustments is required. Assume that the LM310 is ideal and has a zero offset voltage, but the LM311 selected for the circuit has a worst case offset voltage of +10 mV. If V_{in} and V_{out} were exactly the same value, the LM311 would say that V_{out} was less than V_{in} all the time. In this situation the system would always track the input signal at the TRACK command.

To achieve the minimum tracking function the unity gain inverter of Figure A7 was placed in front of another maximum tracking circuit. The output of this minimum tracker is the

ORIGINAL PAGE IS
OF POOR QUALITY

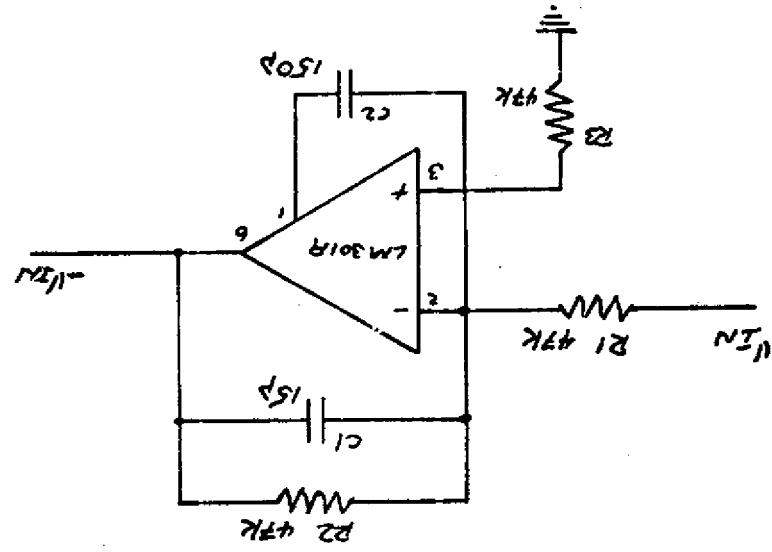


Figure A7 - Unity Gain Inverter

negative of the minimum value of the input voltage. It might seem reasonable to build a minimum tracker by simply reversing the input to the LM311 in the maximum tracking circuit. This method will not work, however, due to the positive input current of the LM310 follower. Since the LM310 has a positive input bias current, the output of the maximum tracker will slowly decay with time while in the hold mode. Consider the situation of a constant input voltage (input to the ZOI system) with a small amount of noise on it. In this situation the maximum tracker would take on the value of some local maximum and begin to decay. When it had decayed to much, it would be reset by the LM311, since it must eventually decay to a level below the signal level. If we simply switched the inputs to the LM311, this circuit would not work as a minimum tracker since it would be decaying away from the value of the signal. Since it will always be below the signal once it start to decay, it will never be reset to a value close to the true signal value.

Amplifier Q3 of Figure A1 computes the average value of V_{\max} and V_{\min} , or the center of the aperture. Since $-V_{\min}$ is available from the minimum tracker, one wishes to compute $0.5(V_{\max} - (-V_{\min}))$. This is accomplished with the circuit of Figure A8, a differentail configuration. If R3 is adjusted so that the ratios $R1/(R2+R3)$ and $R4/R5$ are equal, the the output of the stage is given by,

ORIGINAL PAGE IS
OF POOR QUALITY

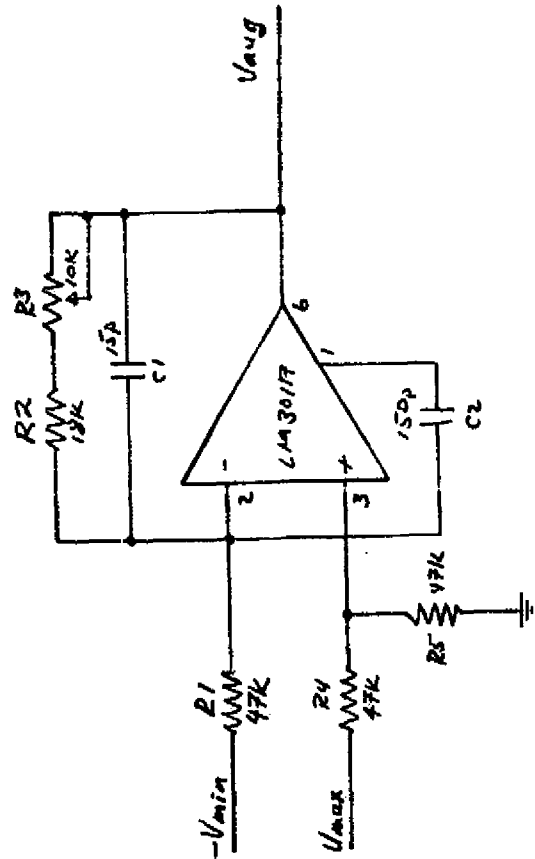


Figure A8 - Voltage Averager Stage

ORIGINAL PAGE IS
OF POOR QUALITY

The ratio $R4/R5 = 0.575$, which is equivalent to a gain change in the signal.

Feedforward compensation is used in the averaging stage to improve the step response of the LM301A. This is done so that the output of the stage may be used in the TS following the change at the inputs.

The error comparator, Q5 of Figure A1, provides the ERROR signal if the aperture width exceeds the threshold V_k . Figure A9 provides a detailed schematic of this circuit. The pertinent LM311 input specifications are;

Input offset voltage	10	MV	(maximum)
Input offset Current	70	nA	(maximum)
Input bias current	300	nA	(maximum).

The low bias currents of the LM311 allow the resistor network of R4 and R5 to be used to compute the difference $V_{\max} - V_{\min}$. Assuming R4 and R5 are driven by ideal voltage sources, the voltage at the junction of R4 and R5 is $0.5(V_{\max} - V_{\min})$. The LM311 compares this with the reference voltage, V_k . R1, R2, and R3 provide a reference voltage over the range 0 to 0.5 volts. C1 bypasses noise at the minus input of the LM311. Referring to Figure A9, note there is a 1.2 K ohm imbalance in the input impedance to the LM311, however, even under worst case conditions, the low input and bias currents of the LM311 hold the maximum error to 0.5 mV. This error can safely be ignored.

The output of the LM311 (pin 7) is an uncommitted collector. This collector is tied to the junction of R7 and R8, R7 and R8

ORIGINAL PAGE IS
OF POOR QUALITY

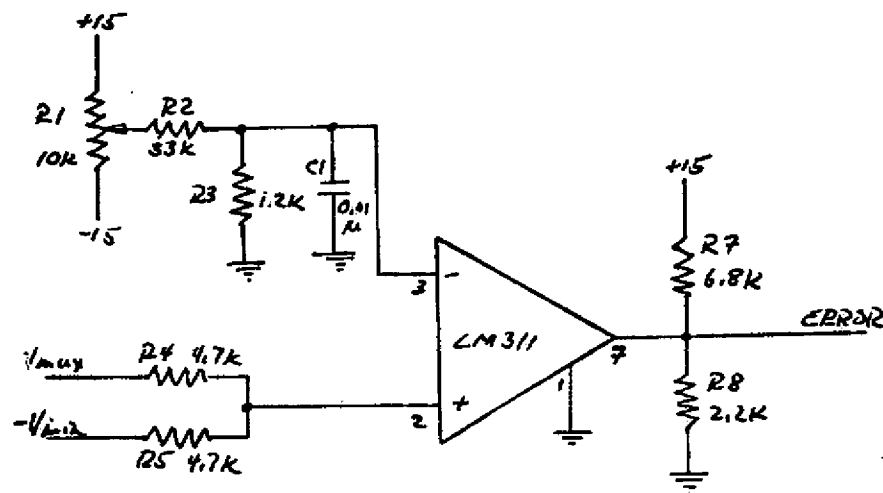


Figure A9- Aperture Width Comparator

form a resistive voltage divider, that limits the range of the ERROR signal to 0 - 5 V, making it compatible with transistor-transistor logic.

Hardware System Output:

Direct output of the hardware system is presented in Figure A10. Figure A10-a shows the output for a triangle wave input of low frequency. It is stressed that this is the direct output of the system, not the reconstructed output. A device with memory is required to generate the reconstructed signal representation, and the hardware system described in this appendix has no such memory.

The steps in the direct output occur with the OUTPUT SAMPLE command. Thus a given segment in these figures provides amplitude information for the previous segment length.

Figure A10-b provides an example of the system output with a triangle wave of increasing amplitude. This gives the reader an idea of the nature of the representation for various amplitude signals, relative to the aperture width.

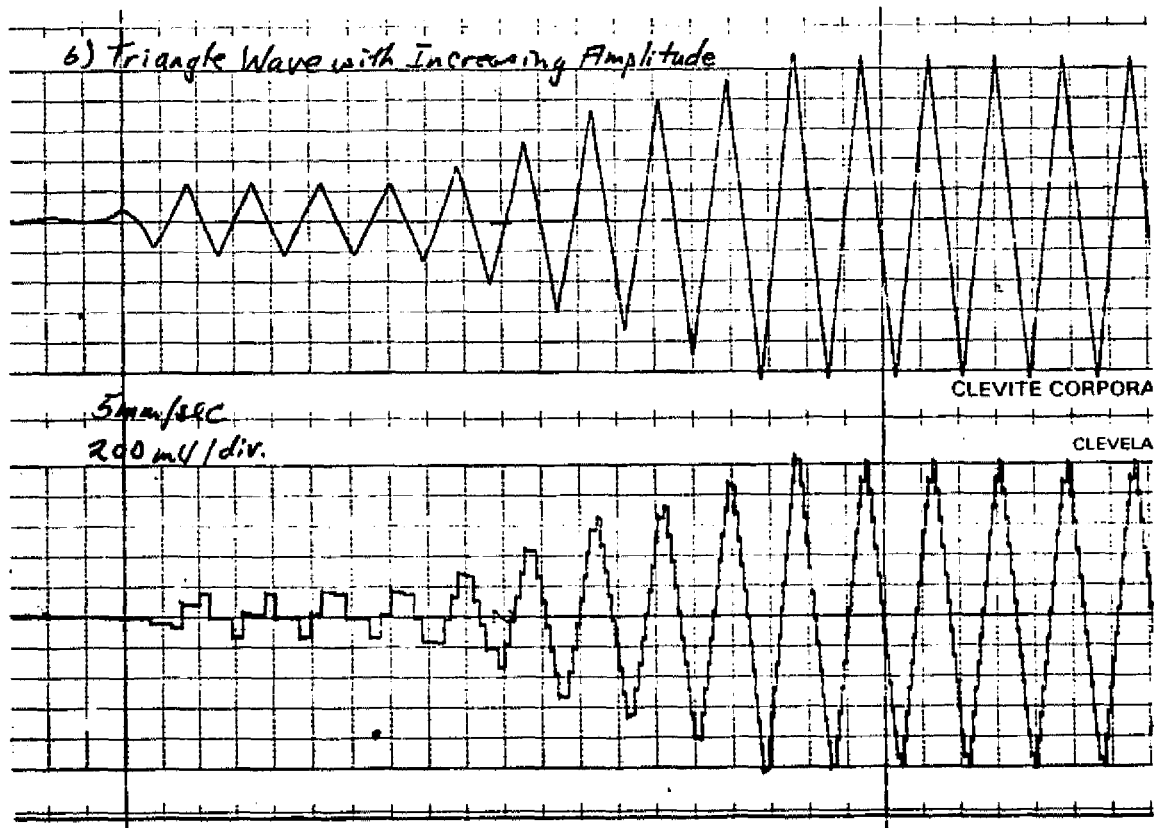
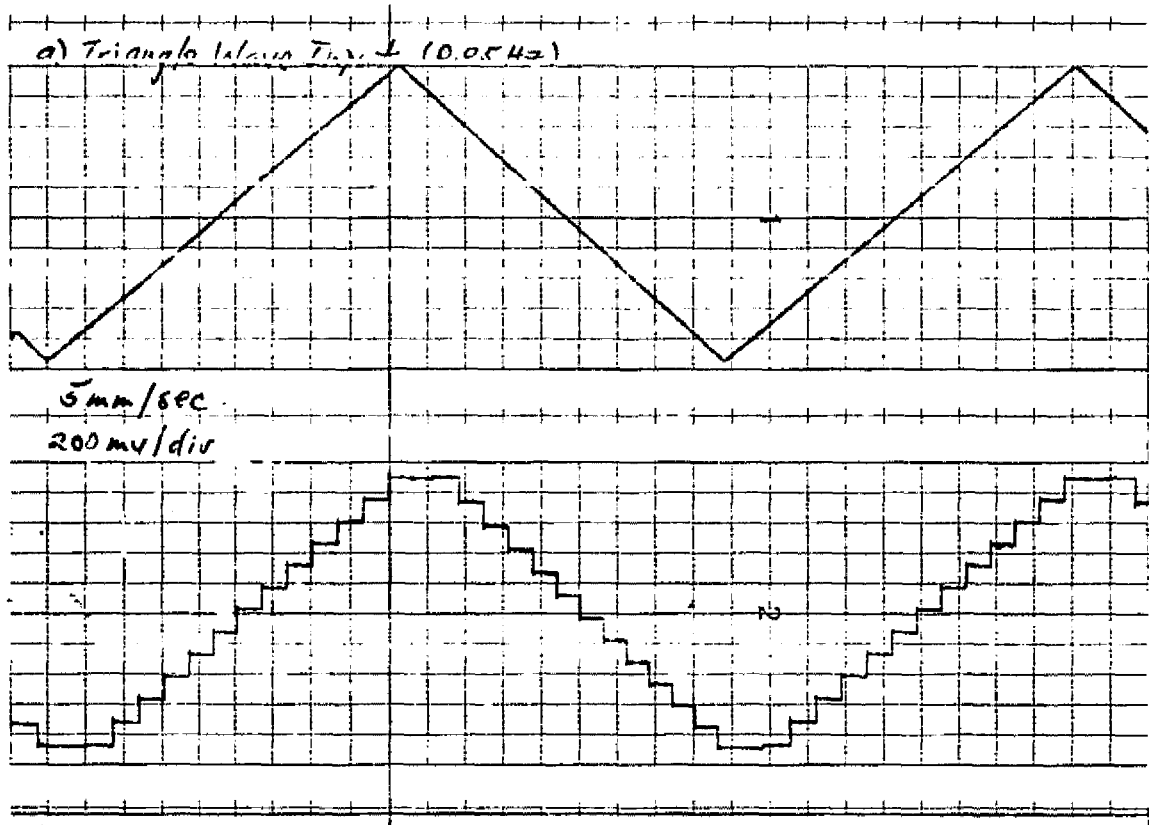


Figure A10 - Examples of Direct System Output

BIBLIOGRAPHY

- 1) C. A. Andrews, J. M. Davies and G. R. Schwarz, "Adaptive Data Compression," Proc. of the IEEE, vol. 55, pp. 267-277; March, 1967
- 2) Corday and D.W. Irving, Disturbances of Heart Rate, Rhythm and Conduction, W. B. Saunders Comp., Philadelphia, 1962
- 3) J. R. Cox, F.M. Nolle, H.A. Fozzard and G.C. Oliver, Jr; "AZTEC", IEEE Trans. Bio-Med. Engr., Vol. 15, p 128, 1968
- 4) W. English, How to Use the Nova Computers, Data General Corporation, 1971
- 5) B. Gold and C.M. Rader, Digital Processing of Signals, McGraw-Hill Book Comp., New York, 1969
- 6) L. Ehrman, "Analysis of Some Redundance Removal Bandwidth Compression Techniques," Proc. of the IEEE, vol. 55, pp. 278-287; March, 1967
- 7) J. G. Kemeny and T. E. Kurtz, BASIC Programing, John Wiley and Sons, Inc.; New York, 1967
- 8) C. M. Kortman, "Redundancy Reduction - A practical Method of Data Compression," Proc. of the IEEE, vol. 55, pp 253-263; March, 1967
- 9) B. P. Lathi, Signals, Systems and Communication, John Wiley and Sons, Inc.; New York, 1965
- 10) J. Macy, Jr; "Analog-Digital Conversion Systems," in Computers in Biomedical Research, R. W. Stacy and B. D. Waxman (editors), Academic Press, New York, 1965
- 11) T. Y. Young and W. H. Huggins, "On the Representation of Electrocardiograms," IEEE Trans. on Bio-Med. Engr., pp 86-95; July, 1963
- 12) T. Y. Young and W. H. Huggins, "Computer Analysis of Electocardiograms Using a Linear Regression Technique," IEEE Trans. on Bio-Med. Engr., pp. 60-67; July, 1964

HIGH SPEED EVALUATION OF MAGNETIC TAPE
RECORDINGS OF ELECTROCARDIOGRAMS

by

PETER BARRY KURNIK

Submitted in Partial Fulfillment
of the Requirements for the
Degree of Bachelor of Science

at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June, 1973

Signature of Author. *Peter Barry Kurnik*
Department of Electrical Engineering, May 11, 1973

Certified by . . . *Stephen K. Burns* . . . Thesis Supervisor

Accepted by. . . *David Adler* . . .
Chairman, Departmental Committee on Theses

PRECEDING PAGE BLANK NOT FILMED

ABSTRACT

A system to analyze magnetic tapes of electrocardiograms (ECG's) recorded over 24 hour intervals has been designed. Tapes are examined on a beat-to-beat basis for potentially significant arrhythmias at a rate substantially faster than required to acquire the data.

Using a Data General Nova computer and a special design interface system, the magnetic tape recordings are sampled 200 times per second (real-time). This amount of data is reduced by a factor of 10 by creating a horizontal line segment approximation. This is followed by a block which detects the QRS complexes, another which measures the width and magnitude of the QRS complex plus the baseline and QQ interval, and another block which uses these measurements to decide if the complex represents a PVC or a normal heartbeat. Output consists of coded representations of the width, magnitude, and QQ interval, plus a flag marking detected PVC's.

This system has been tested on several tape recordings of clinical data, containing over 75 premature ventricular contractions (PVC's). Every PVC was flagged. The rate of false positives is under 1%, with the expectation of being greatly reduced. The program runs at 8 times real-time, and calculations indicate that it presently should be able to run 14 times real-time on the Nova. It is expected that the same program should run 60 times real-time on a Nova 800. Reducing the sampling rate to 120 samples per second is probably feasible, which would yield a corresponding increase in speed.

ORIGINAL PAGE IS
OF POOR QUALITY

ACKNOWLEDGEMENTS

I thank Professor Stephen K. Burns very much for all his advice and guidance, and for providing the facilities for this project.

Special thanks are given to Joe B. Walters for his suggestions on how to attack the problem and his omnipresence when assistance was needed.

TABLE OF CONTENTS

Abstract	2
List of Figures	5
Background, and the Problem to be Solved	7
The Normal Heart	7
Arrhythmias	9
Clinical Significance--Research	13
Previous Work	15
The Solution	17
Requirements	17
Processing Required	19
The Implementation	25
Suggestions for the Future	84
Conclusion	86
References	88

LIST OF FIGURES

Figure 1:	Physiology of the heart	8
Figure 2:	Normal ECG waveform	10
Figure 3:	Three PVC's	11
Figure 4:	Flowchart of master algorithm	26
Figure 5:	System hardware interconnection	27
Figure 6:	Flowchart of Zero Order Interpolator	30
Figure 7:	Interpolation with default parameters	31
Figure 8:	Interpolation with aperture twice normal	33
Figure 9:	Interpolation with aperture four times normal	35
Figure 10:	Interpolation with normal aperture, at 120 samples per second	37
Figure 11:	Interpolation with aperture twice normal, at 120 samples per second	39
Figure 12:	Interpolation of a noisier recording, default parameters	41
Figure 13:	Interpolation of a noisier recording, with aperture twice normal	43
Figure 14:	Interpolation of a PVC, with aperture twice normal	45
Figure 15:	Interpolation of a second PVC, default parameters	47
Figure 16:	Interpolation of a third PVC, default parameters	49
Figure 17:	Flowchart of detection algorithm	53
Figure 18:	A detected PVC	58
Figure 19:	Another detected PVC	60

Figure 20:	Two more detected PVC's	62
Figure 21:	A longer recording	64
Figure 22:	PVC's from another patient	66
Figure 23:	A direct recording at 8 times real-time	68
Figure 24:	An indirect recording at 8 times real-time	70
Figure 25:	A false positive identification	72
Figure 26:	A second false positive identification	74
Figure 27:	A third false positive identification, and its solution	76
Figure 28:	Flowchart of display algorithm	79

BACKGROUND, AND THE PROBLEM TO BE SOLVED

The electrocardiogram (ECG) is the potential created on the surface of the body by the electrical activity of the heart. Because many problems with the heart are involved with a defect in its electrical generation or conduction apparatus, the ECG is a very useful clinical tool. "No cardiac study is complete without an electrocardiogram. No electrocardiographic analysis is more detailed than a computer aided one."²⁰

The Normal Heart

The heart is composed primarily of several layers of muscle. When they contract in orderly fashion, blood is circulated to the entire body. The rhythmic contraction is controlled by a specialized electric generation and conduction system in the heart, as seen in figure 1. An impulse is started each beat in a natural pacemaker in the right atrium called the sinoatrial node. The impulse travels through the atrium, depolarizing it and causing it to contract, and stimulates the atrioventricular node. It continues along a comparatively high speed system called the Purkinje fibers, spreading through both ventricles and causing them to depolarize and contract. After a short period of time,

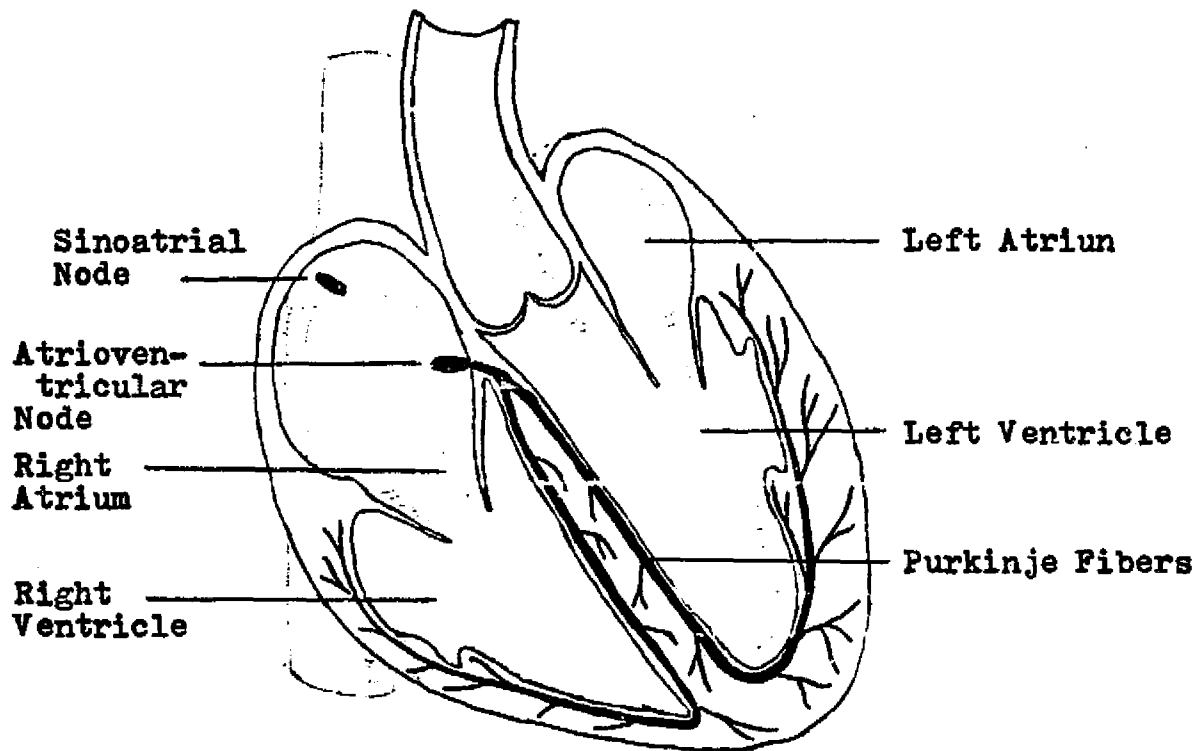


Figure 1

**Physiology of the heart; including pacemaker
and conduction system**

the heart repolarizes and is ready for another cycle.

The movement of the electrical impulse through the heart can be reasonably modelled as an electric dipole⁶. It obviously has a three-dimensional pattern of movement. This equivalent dipole is observed on the body surface by two or more electrodes, so that the placement of the electrodes determines which planar projection of the dipole is observed. The observed waveform is dependent on skin resistance, amount of body fat, heart placement, and other factors in addition to electrode placement, so that it becomes important to locate the electrodes skillfully to obtain a useful ECG. Figure 2 depicts a stereotyped ECG. The P-wave corresponds to the atrial depolarization, the QRS complex to the ventricular depolarization, and the T-wave to ventricular repolarization.

Arrhythmias

Problems with the heart's electrical system may be due to the impulse origin or to abnormalities of the pathway, and may be either regular or intermittent. Many classifications have been described, and one particular one is the premature ventricular contraction (PVC).

In a PVC, the focus, or impulse origin, is not in the sinoatrial node, but in a ventricle. It is earlier in a cycle than the normal impulse would be expected, and, therefore, overrides it. It is generally unable to make use of the Purkinje fiber conduction system, and consequently travels slower and creates a wider QRS complex. Figure 3 shows three

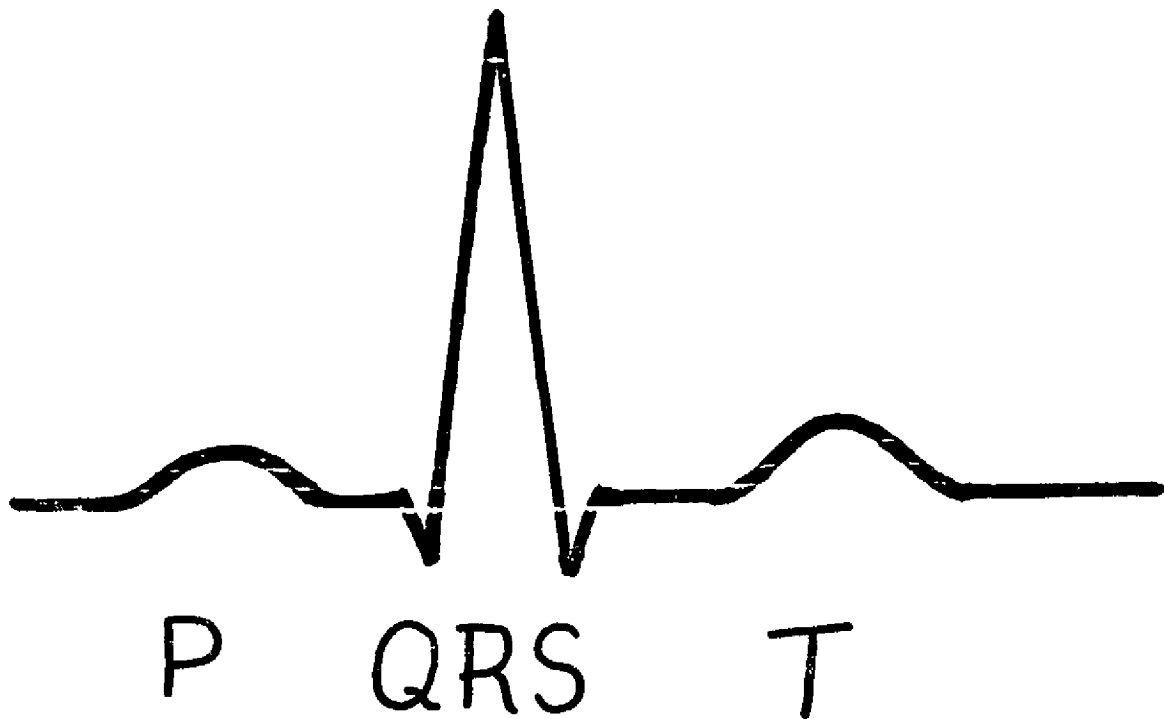


Figure 2

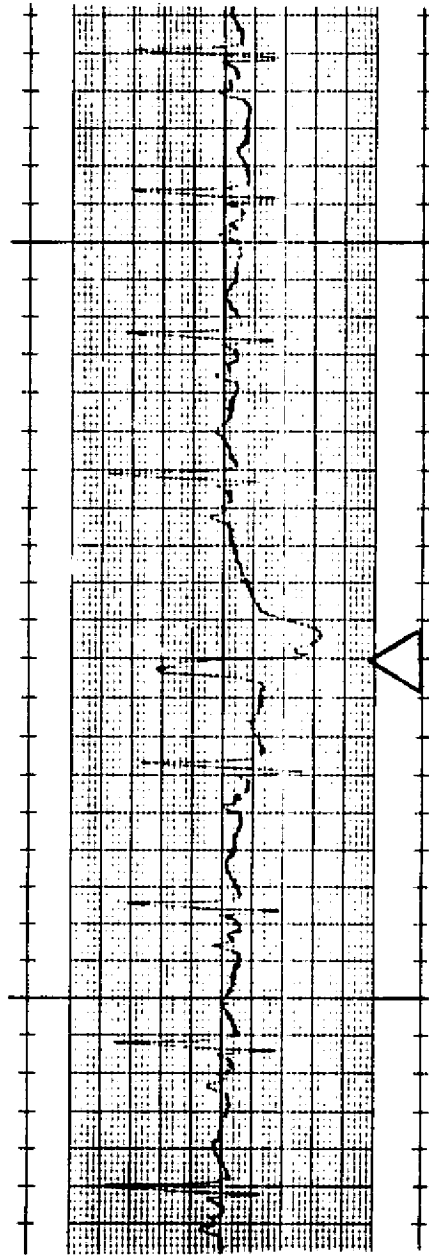
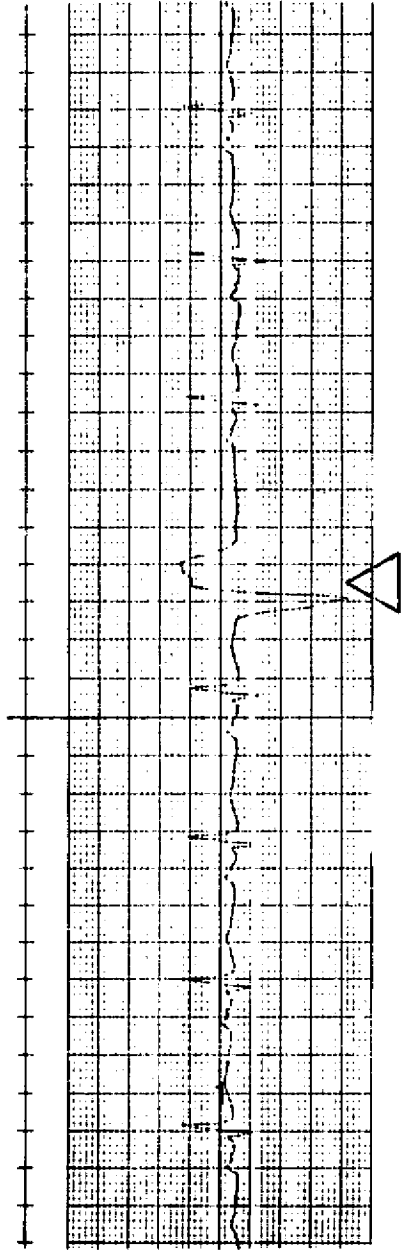
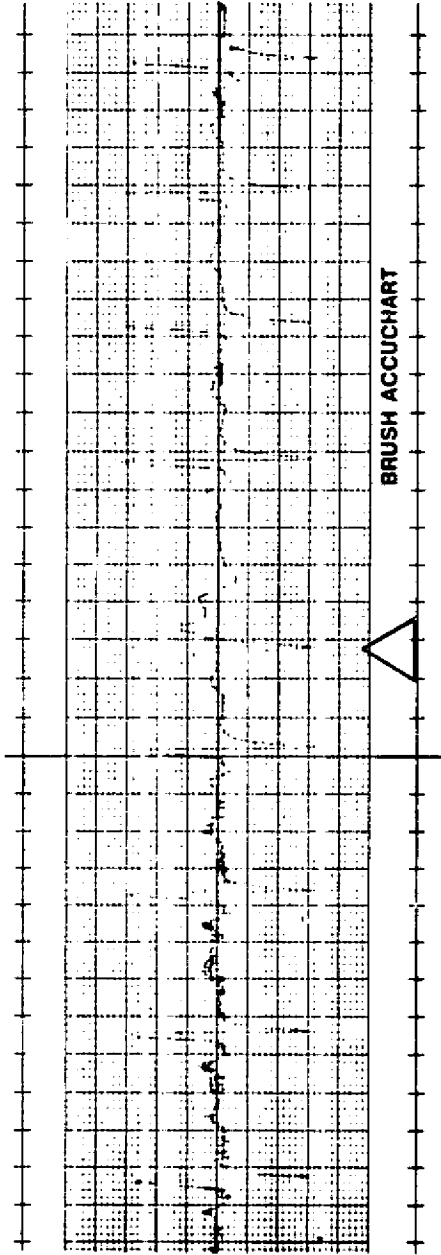
Normal ECG waveform

P-wave: atrial depolarization

QRS complex: ventricular depolarization

T-wave: ventricular repolarization

Figure 3: Three PVC's, illustrated in the context of their normal heartbeats. Note the prematurity in all of them and the varying degree of transformation of the QRS morphology.



different PVC's in the context of beats which are normal for that patient. If all PVC's have the same ventricular focus, they will tend to have the same shape. If the PVC's are multi-focal, they may have widely varying morphology, and this is symptomatic of different problems than unifocal PVC's. The PVC is always early, usually has a compensatory pause following it, is usually wide, is often taller than normals and inverted in polarity, and may have a bizarre shape.

Clinical Significance--Research

Isolated PVC's may occur in otherwise normal individuals, and probably have little significance as long as they are infrequent. As they become more frequent, and if they are multi-focal, the danger is increased. In particular, if early PVC's fall on T-waves (the electrically vulnerable period) they are likely to initiate fibrillation,¹¹ a disorganized beating of the heart which has very little effective pumping action. Fibrillation is reversible if caught within about three minutes, else death usually results. Until recently, the patient depended on very fast recognition and response to his fibrillation for his life. Since 1965, in some coronary care units, patients with PVC's are treated with anti-arrhythmic drugs such as lidocaine to suppress PVC's and prevent fibrillation altogether. Prior experience had shown that about 15% of heart attack patients in the hospital developed fibrillation. Since then, incidence of ventricular fibrillation has been held to less than 1%.¹¹

This technique has reduced the mortality rate of those in coronary care units from 30% to 20%, or about 60,000 lives per year. This is certainly sufficient to establish a PVC monitor as an important clinical resource.

In order to do scientific research on such topics as testing anti-arrhythmic drugs, large numbers of patients must be monitored for long intervals. Another research area of critical importance is to attempt to find predisposing symptoms in those who "drop dead" from sudden heart attacks. Towards this end an entire community has their ECG's recorded for a number of years. There is currently a joint U.S.-Soviet program underway to study thousands of people.¹ Here again is a huge mass of data to be analyzed. Because PVC's are generally not regular with respect to time, a large sample of data must always be analyzed. Therefore, a method for making an accurate, high-speed detector of PVC's is urgently needed. Human observers are too scarce, expensive, and inaccurate.

PREVIOUS WORK

There has been a great deal of interest in applying computers to the analysis and monitoring of ECG's, since about 1960. Every researcher seems to have his own special idea, consequently there has been a proliferation of hardware used, detection methods, and decision algorithms. Many have been quite successful for either general analysis or a specialized job they have been designed for. One excellent source that covers the entire field from many perspectives is a 1970 book edited by Cesar A. Caceres and Leonard S. Dreifus.³ Seventy-one authors have contributed articles on: instrumentation, techniques in computer programming, utilization and potential, and research for clinical needs. The book covers both the state-of-the-art and a systematic discussion of needs for the future. There is one paper on monitoring, none on high speed analysis. A 1972 paper by Jerome R. Cox and colleagues has a fine review and discussion of who is doing what in computer analysis of ECG's, EEG's, and blood pressure waves.⁶ His paper is also notable for the 300 references included on the field. Many papers on rhythm monitoring are detailed, but, again, none on high-speed evaluation.

High-speed analysis, even specialized to a single feature such as PVC's, is not described in the literature. One technique is to display magnetic tape recordings at 60 times real-time on an oscilloscope that triggers such that normal waves are

superimposed. Abnormal waves stand out visually to the human operator, and the system may be halted at each PVC to provide a strip chart recording. Heart rate may be monitored as an added feature.⁹ One computer analysis runs as fast as 15 times real-time on a PDP-7.¹² Simple analog devices such as high and low rate alarms may also be used. This author is unaware of other working systems in high-speed evaluation.

The data reduction scheme used in this paper, which is a critical part of the program, is based on a paper by J.B.Walters, Jr.,¹⁹ and is similar to an algorithm used for a similar purpose by Jerome R.Cox, Jr.⁷

THE SOLUTION

Requirements

First, the analysis must be accurate in its decisions. The number of false negatives, PVC's classified as normals, must be very close to zero, and the number of false positives, normals and artifacts classified as abnormals, must be reasonably small. The acceptable number of false positives depends on the respective costs of false negatives and positives, if the analysis is to be used clinically or for research, and on the condition of the recording. On a patient with a heart rate of 100 beats per minute and 1 PVC per minute, a 0.1% false positive rate will have 1 false positive for every 10 true positives and a total of 86 false positives in a 24 hour recording. This would probably be a good level. The number of PVC's is much smaller than the number of normals, and it is they that hold the information that may indicate disease in the patient. Also, if the patient has more than one type of PVC, it is very important not to catch some types and miss others. This is because runs and patterns of PVC's, and the realization that they are multi-focal, can be as critical as finding individual PVC's. Doctors may also be particularly prone to doubting the validity of a program that misses an arrhythmia that they spot visually, and without the physicians cooperation, no automation will be used. Hours of accurate detection of normal QRS complexes may be less important than the identification of a few anomalous beats that possibly

warn of more severe arrhythmias to come."⁴ Three other computer systems that were looking at arrhythmias respectively report finding 71% of PVC's,⁸ 95% of PVC's,¹² and 80% of PVC's and AFC's.¹³ It is also important to minimize the number of false positives in order to preserve the usefulness of the analysis, but this error is not as critical.

The second requirement is that the program be fast. Sixty times real-time is a useful speed, and a goal that has been talked about. Because of the peculiarities of PVC's and the nature of the studies that involve PVC detection, huge amounts of data are generated. As mentioned earlier, there are likely to be very large numbers of 24 hour tapes that will need to be analyzed. Real-time analysis would be uneconomical in equipment tied up, personnel required, and time needed before output data was available. Because a complete analysis is not needed, only a PVC detection, there should be considerable room for speeding up the process. One computer arrhythmia detector has run as fast as 4 times real-time on a PDP-4 and 15 times on a PDP-7.¹²

Another requirement for a successful implementation is that it be dependable and easy to use. The personnel using the system are likely to be non-technical and/or non-medical. Therefore the set-up, the switching from one tape to another, the output, and the record-keeping must all be designed to be handled in a simple routine. Any parameters that need adjusting, including speed changes, must be easily accessible and held to minimum. The program itself should

handle any abnormalities that are at all likely to occur, and when it is unable to handle some problem it should provide an indication to the operator of the difficulty.

The final general requirement for a good system is that it be flexible. Not only must it be able to run at different speeds, but it must be adjustable to handle widely varying, including noisy, data. As more data is gathered and analyzed, the parameters should be adjustable for optimum discrimination. A new algorithm for a module such as the detector block should be easily switched in, and the overall program should be expandable to include features such as statistical analysis. So, a successful system must be a fast, accurate, flexible, easy-to-use method of detecting premature ventricular contractions.

Processing Required

There are six main blocks that nearly any solution would have to encompass. In order, they are input, data reduction, detection, measurement, decision, and output.

Input

In order to store the ECG in the computer, the analog signal must be sampled and converted to digital form. A variety of methods and sampling rates, and a variety of reasons to support them, are available in the literature. The American Heart Association says that 500 samples per second(sps) and a 9-bit digital output are needed to fully analyze an ECG.⁶ Some groups agree with this rate,^{4,7} while others feel 200 sps is sufficient,^{9,16} and others feel that even 120¹⁷ or 125 sps¹² is all that is needed. Not as much

detail is required for a PVC detector as for a full examination of the complete ECG, so a lower sampling is probably permissible.

Data Reduction

Any sampling rate comparable to those indicated above provides too many points to be handled at very high speeds. For example, 200 sps real-time extrapolated to 60 times real-time would require 12,000 individual points per second. Eighty nsec. is far from enough time to average on each piece of information. Therefore, the amount of data must be reduced, but without losing important information in the waveform. Between QRS's, and that is the vast majority of the time, the signal is quite flat and with little information value. A good reduction would preserve a QRS in its entirety and describe the baseline with only a few numbers. One algorithm that does this well is the AZTEC preprocessor developed by a group at Washington University.^{5,7} That program approximates an analog signal with a series of horizontal line segments and slopes. As long as the input signal stays within a preset aperture of the last output value, the output stays constant (horizontal). When the threshold is exceeded, a new line segment is started and the process repeated. Consecutive short segments, i.e. steep parts of the wave, are further approximated by a single sloping line. The size of the aperture regulates the amount of data reduction and the amount of detail preserved. Low amplitude variations are lost, while steep slopes are preserved almost completely.

The authors recommend a window of $\frac{1}{4}$ of a normal QRS height. They estimate an average data reduction on ECG's of 10:1 with little loss of information. A similar preprocessor, which involves an approximation using only horizontal line segments, is called a Zero Order Interpolator and detailed in another source.¹⁹ This is faster, preserves steep segments more faithfully, and has a larger number of output segments than AZTEC. The data storage is also simpler. Using this interpolated data, an analysis should be much swifter.

Detection

First, it must be decided what one is looking for and then it must be decided how best to look for it. For a PVC analysis, the QRS complexes provide sufficient information. The problem at this stage is then to find all QRS's, normal and abnormal, and reject everything else. A variety of techniques have been employed, but they all center on the two main distinguishing features of the QRS, its large amplitude and large slopes. Some find the beginning of the QRS simply by finding a slope of sufficient steepness,^{2, 12, 16} another requires a minimum slope to exist for a minimum length of time,²⁰ still another requires two slopes that collectively exceed both critical positive and negative slopes within 150 msec. to define a QRS,¹⁵ while others make their detection a combination of sufficient slope and sufficient magnitude.^{8, 9} In addition to finding all true events, such noise problems as steep spikes and baseline shifts should

ORIGINAL PAGE IS
OF POOR QUALITY

be taken into account as much as possible and this presence should lead to a minimum of false detections. This stage is intimately related to the next, that of measurement.

Measurement

Once the QRS complexes have been found, their critical parameters must be calculated for use in deciding if it is a PVC. P-waves are not used since they are difficult to measure (amplitude typically only twice that of noise²⁰) and they are of secondary importance in finding PVC's. Measurements are usually made of the magnitude and width of the QRS, the baseline, and the rate or interval between complexes. Within this agreement, there is once more a wide choice of methods. Some investigators determine onset and end of the complex by looking for a flat stretch of a certain length^{5,20} or a stretch which has a slope less than some cutoff,¹⁶ while others invoke procedures utilizing first and second derivatives.^{12,20} The magnitude measurement may be either a signed number in relation to the baseline or a peak-to-peak amplitude. Interval is self-explanatory, but can be chosen to be QQ, RR, or some other regular interval. Baseline, while not used in deciding on abnormality, is used as an intermediate measurement to provide a local reference zero instead of an arbitrary constant. It can be measured as the longest flat stretch, as the magnitude at some particular temporal location in reference to the QRS, or as the most common value in a histogram plot of the total waveform.²⁰ This is a very critical part of the

program. An error in measurement can easily yield values that would classify a normal as abnormal, or vice versa. Low and high amplitude and low(baseline shift) and high frequency (spikes) noises all contribute their vagaries to the problem. Peak-to-peak magnitude is an easy measurement, but onset and end of the QRS and a good baseline are often obscured because there may be no significant stretches of flat wave at the right places.

Decision

The decision module is in a sense the heart of the analysis, where all the other data is gathered and analyzed. This project was defined to search only for PVC's, so this algorithm may be comparatively simple. However, in addition to deciding what is a PVC, families of PVC's having the same focus and, therefore, similar shape can be found and runs or other patterns of PVC's can be discovered. Features of PVC's which could be used to distinguish them include prematurity, compensatory pause, width, height, bizarre shape of QRS, and polarity. The two most widely used are width, which one source⁶ says is the most important number, and prematurity. Definitions of prematurity range through 60%, 70%, 80%, 85%, and 90% of normal interval.^{8,9,13} The normal is generally a running average of the normal intervals. This is essential since not only does normal vary greatly from patient to patient, but also greatly for one patient over a period of time as long as 24 hours. Some of the other features mentioned are also used by some researchers.

In general, the parameters have been considered independently, but it is also feasible to accept a certain degree of simultaneous variation in each of a number of parameters to constitute a basis for abnormality. Normally, once PVC's have been found, they would be flagged in some way and perhaps processed further in some statistical survey.

Output

There is no particular output which is best in general, instead it depends on the specific application. The only necessity is that it provide a means of accessing the abnormals and a coding of why they were considered abnormal. The easiest output is to simply supply a running string of the parameter measurements and the flags of PVC's. Unless delay of some sort is utilized the parameters and flags will necessarily appear at the output later than the raw signal. This presents no difficulty if it is feasible, in the follow-up stage, to play the output tape backwards and hence see the flag before the PVC. At high speeds it would be difficult to provide hard copy output of abnormals in a single pass, simply because a strip-chart recorder cannot move that fast. It would presumably be not too difficult to build a piece of hardware for the second stage, which accepts a magnetic tape recording of the ECG, coded parameters, and flags. A typical option would be to play this tape backwards, turn on a strip chart recorder when a flag is detected, output that PVC and as much surrounding context as desired, turn the strip chart off, and continue until another PVC is

found. If all that is required is a statistical summary or histogram of abnormal events, that can of course be provided with only one pass in whatever format the user desires. Another idea which has been suggested is to make use of the digital formatting, and store PVC's and data on digital storage of some kind. A playback of this could be done much more flexibly and without a tape recorder.

The Implementation

A system which performs the tasks described earlier has been established and is working. The actual development has been done on a Data General Nova computer. It is intended that the present program be transferred without change to a Nova 800 in order to utilize better the speed potential of the system. Most instructions on the Nova require 5 to 6 usec. for execution, so that an immediate gain in speed of a factor of 4 or 5 could be expected by using the faster Nova 800. The Nova is a four-accumulator, 16-bit machine. The entire program has been written in assembly language, and it currently contains about 500 executable instructions. Assembly language was used because of the critical speed requirements of the program. A flow chart showing the entire program in rough blocks is in figure 4.

The program was developed under a disk operating system, permanent storage is on Linc tapes. Figure 5 illustrates the full implementation. Communication between the computer and the outside world is done through an Analog Input and an Analog Output, which are parts of an Analog Chassis that

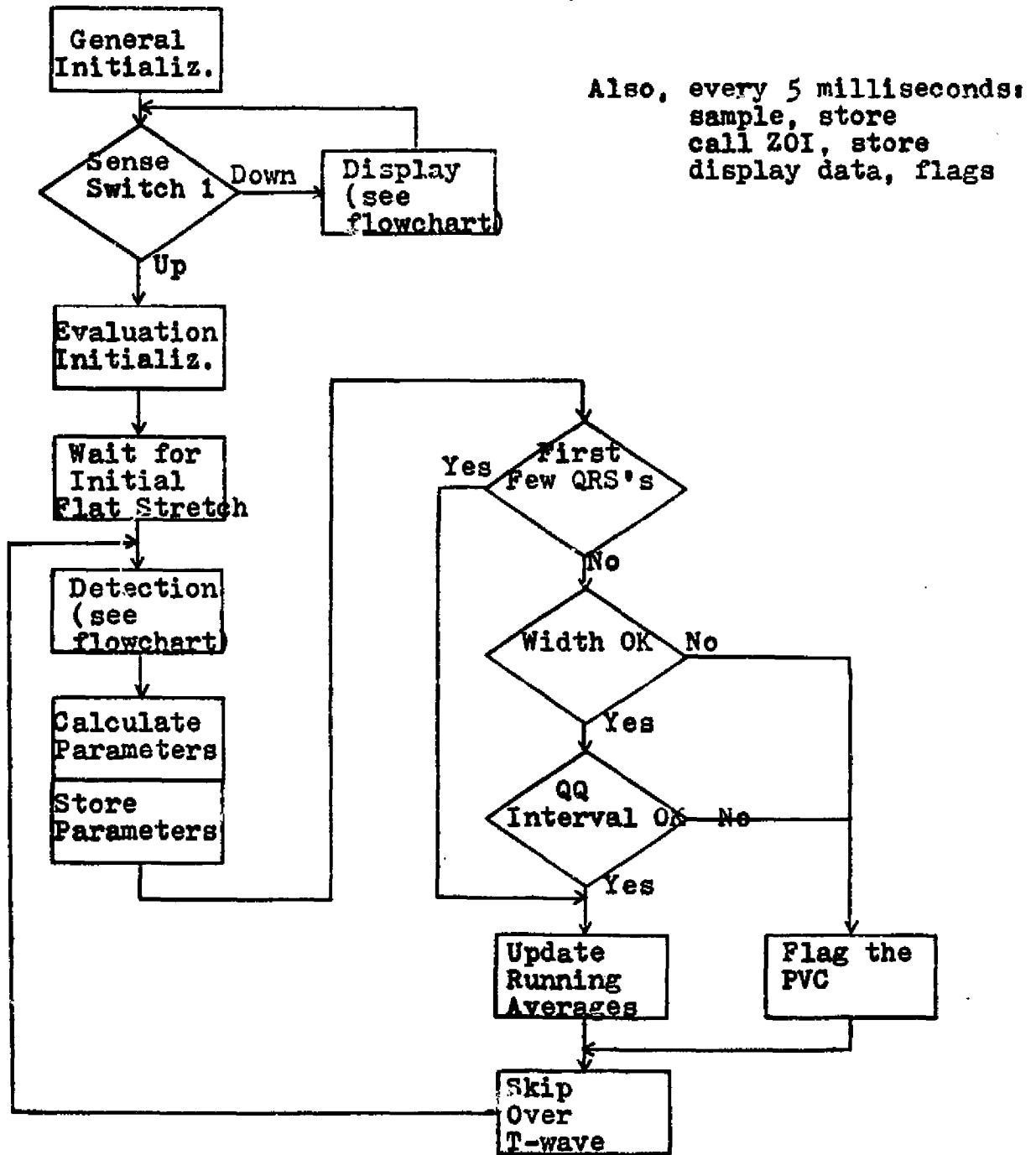


Figure 4
Flowchart of master algorithm

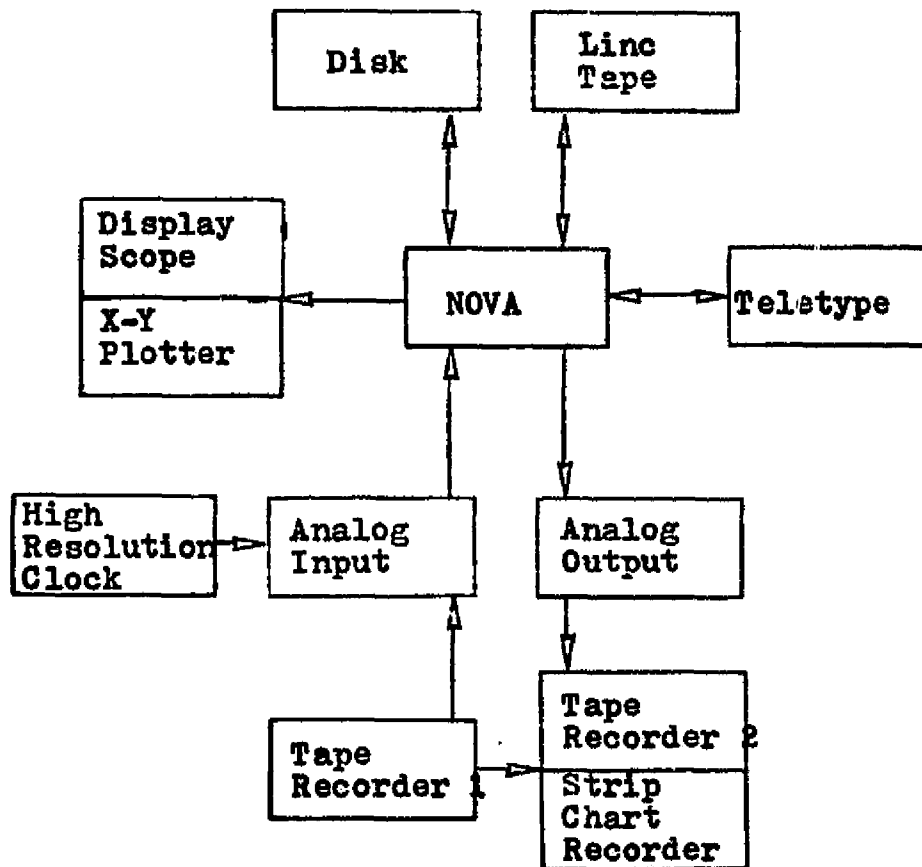


Figure 5

System hardware interconnection

was designed to be used in a clinical computation system.¹⁸ They provide, respectively, the buffered analog/digital and digital/analog converters. In the same Chassis is a High-Resolution Clock which is program adjustable to provide accurate sampling of the signal for any selection of processing rate. Communication between user and program is via teletype under DOS control. A tape recorder is needed for the input signal and the same or different tape recorder and/or strip chart recorder are needed to receive the output. It is highly recommended that a display scope be provided to ascertain that gain and offset on the input are appropriate, and special signals are provided for an X-Y plotter if that is desired.

Input

The ECG signal from a tape recorder is delivered directly to the Analog Input(AIN) mentioned above. The full input range of the AIN is adjustable from ± 0.2 volts to ± 4 volts, and a DC offset of ± 5 volts provided at the front panel. When triggered by the High-Resolution Clock(HRC) described immediately below, a 10-bit analog/digital conversion is made. The period of the HRC can be adjusted by the computer in steps of 1 μ sec. A 200 sps rate has been used for real-time analysis, so that the HRC provides a conversion command to the AIN every 5000 μ sec. As the processing rate is increased the HRC period is decreased accordingly, by changing the value of a single variable in the program. Every time the AIN completes an input, an

interrupt handler is called in the program. The sample of data is stored in a circular buffer that is long enough to store a signal that will fill the display screen. At this time, the data reduction algorithm also considers these points, as described following.

Data Reduction

The method of data reduction is the zero order interpolator (ZOI) described earlier. Basically, a horizontal line segment approximation is made to the input signal. Whenever the signal differs from the most recent output by more than a prescribed aperture, a new output segment is begun. Alternately, if the output remains constant for greater than a prescribed length of time, a new line is started in order to provide a reasonably regular data stream. A flowchart of this structure is in figure 6. The line segments are stored as word pairs, representing the amplitude and length of the segments, in a circular buffer which is again sufficiently long to provide a full-screen display and to temporarily buffer heart-rate changes and lags in the analysis. The size of the aperture may greatly affect the shape of an interpolated signal. A larger window will ignore low amplitude deviations, but preserve large ones, in addition to providing a larger data reduction. Figures 7 through 16 illustrate the properties of the ZOI and the effect of changing the aperture. Also illustrated is the effect of changing the sampling rate from 200 sps to 120 sps. Three PVC's are included for reference.

Sample: supplied by calling routine
 Vmax: running record of local max.
 Vmin: running record of local min.
 Len: running record of local length
 Mlen: max. allowable segment length
 Aldev: max. allowable voltage difference between segments

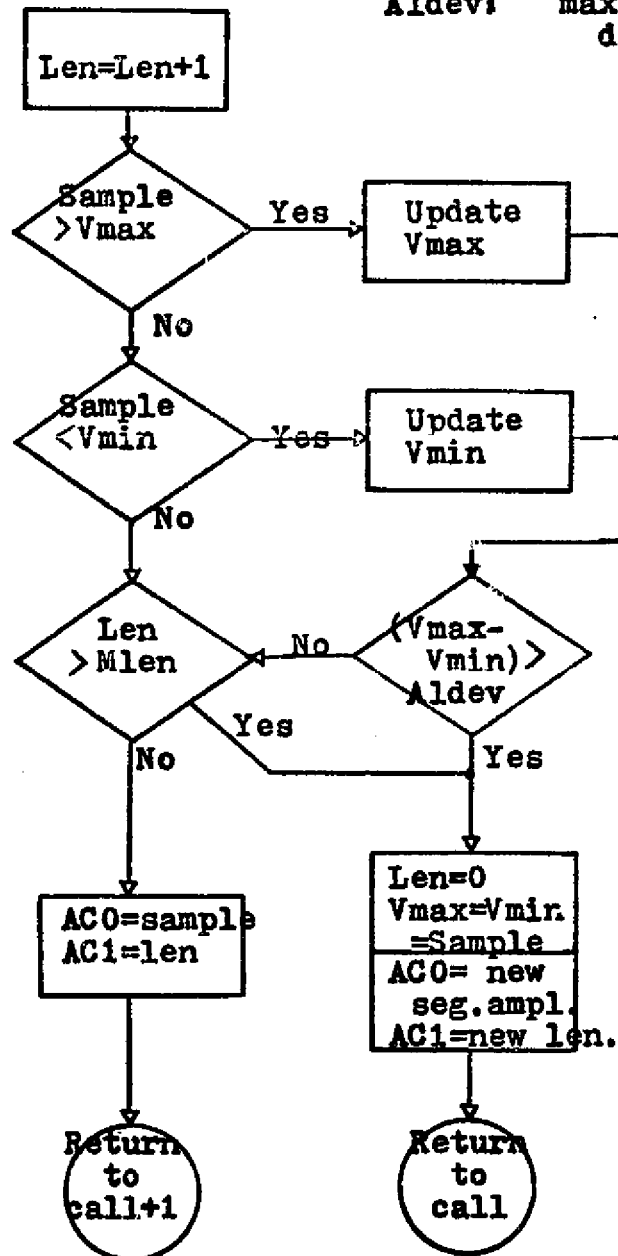


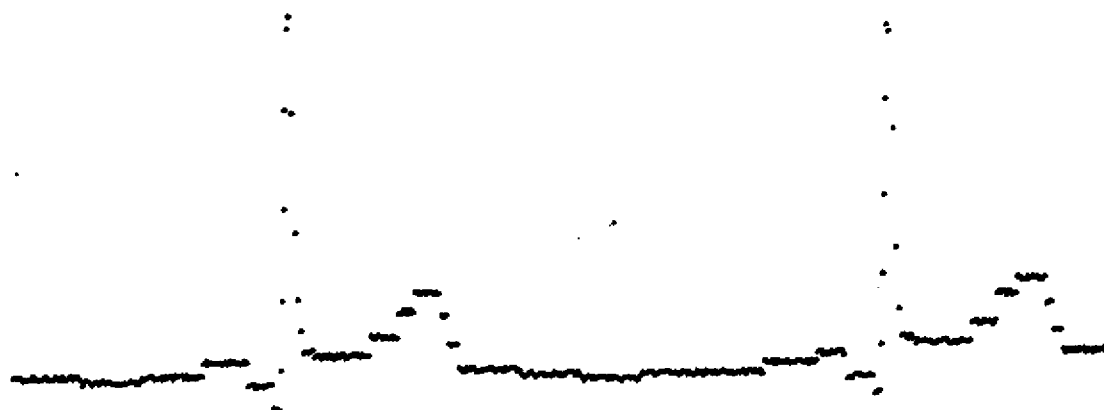
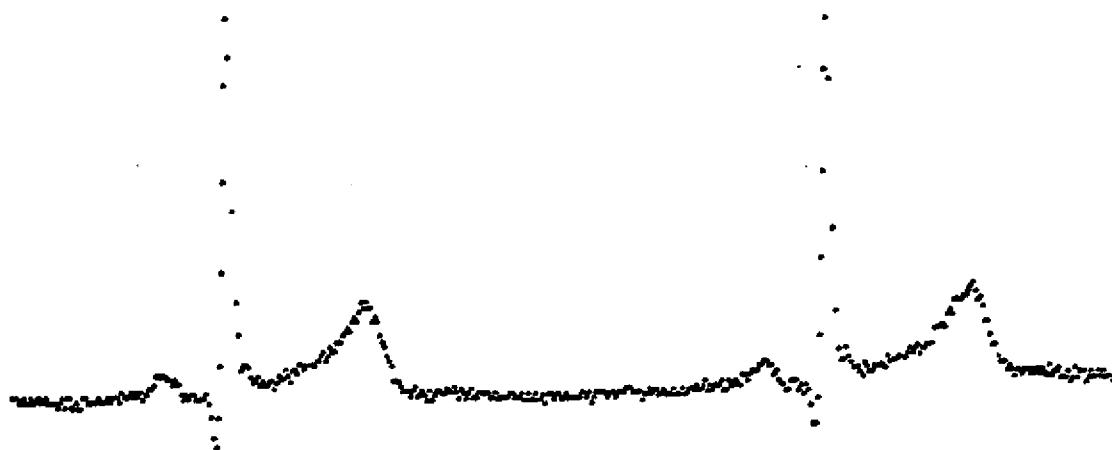
Figure 6

Flowchart of Zero Order Interpolator

Figures 7 through 16 were all produced on an X-Y plotter, using a feature of this program. The top line is the sampled and digitized signal and the bottom line is the output of the zero order interpolator. There are 512 points per line, so that, at 200 samples per second, this represents 2.56 seconds. This is exactly the same picture as seen by the operator on the display scope. Unless specified otherwise, sampling rate was 200 times per second and the aperture was 32 units out of a possible 1024. The normal QRS was adjusted to occupy approximately 512 units, so that the aperture represents about 6% of a normal QRS height.

Figure 7: Interpolation with default parameters.

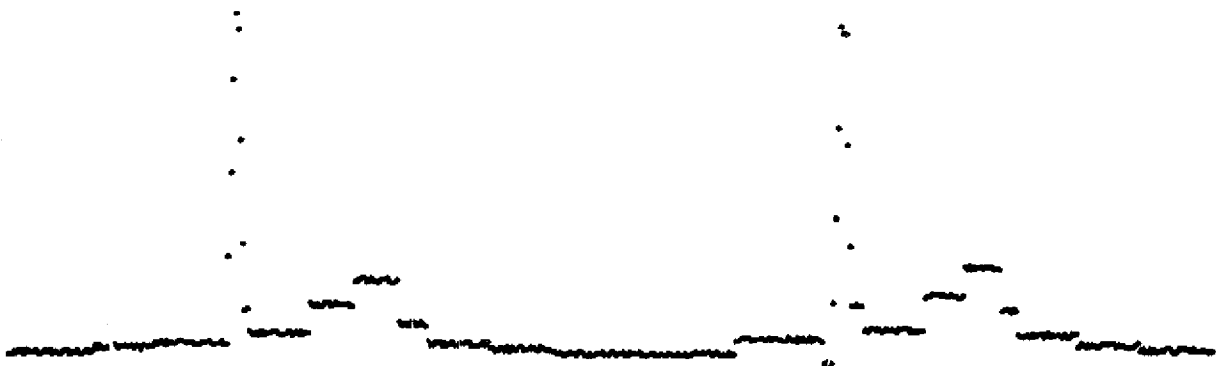
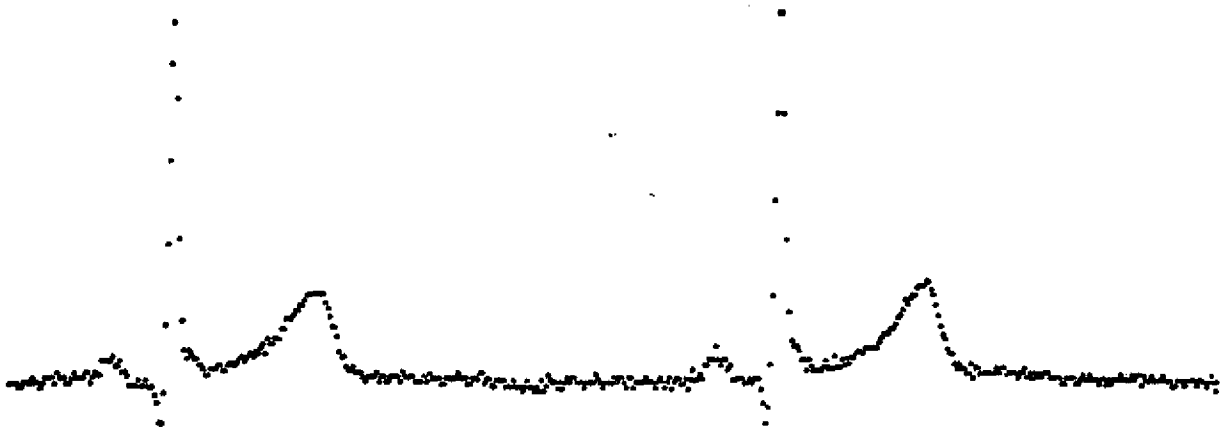
Note that the QRS is virtually unchanged by the approximation, while the baseline and smaller waves have the number of segments representing them greatly reduced. It can be easily verified that the number of line segments is about one tenth the number of original bits.



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 8: Interpolation with aperture twice normal.

This is the same waveform as figure 7, with the aperture now set at about 12% of the QRS height. The P-wave has been lost, the T-wave is still easily recognizable, and, except for the very onset, the QRS complex is still nearly the same as the sampled complex.



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 9: Interpolation with aperture 4 times normal.
This is again the same waveform as in figures 7 and 8, with the aperture now set to 128 units. The T-wave has lost its morphology, although its location is well marked. The QRS is still very well represented. The data reduction is now about 15:1.

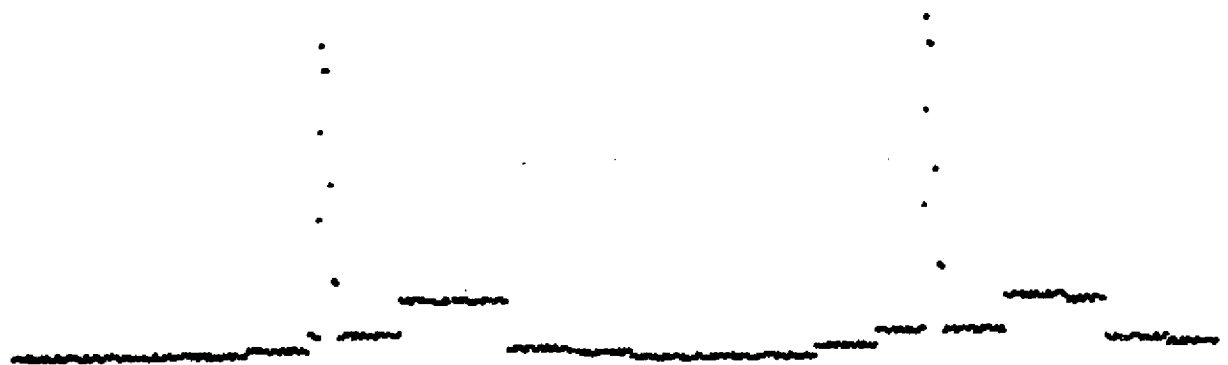
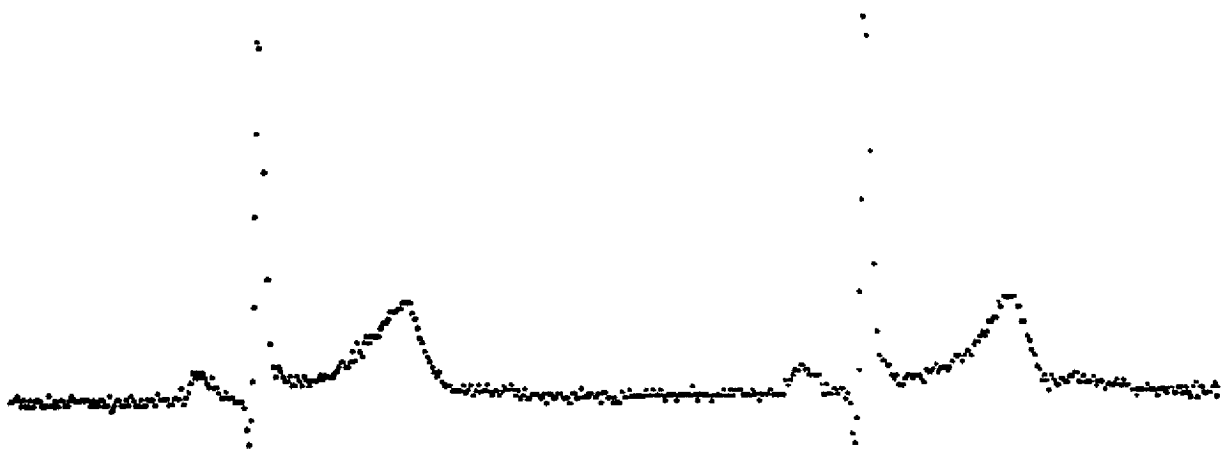


Figure 10: Interpolation with normal aperture, at 120 samples per second. This is the same waveform and aperture as figure 7, with $3/5$ the number of sampling points per second. The width of the picture is now 4.27 seconds. In the upper trace, the amount of detail is noticeably less than at 200 samples per second, but probably quite sufficient for analysis. Again, the interpolation at this aperture provides an excellent reproduction of the QRS.



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 11: Interpolation with aperture twice normal, at 120 samples per second. This corresponds with figure 8 in its data reduction and feature representation. The P-wave is gone, the T-wave detectable, and the QRS totally reproduced with the exception of the tip of the Q-wave.

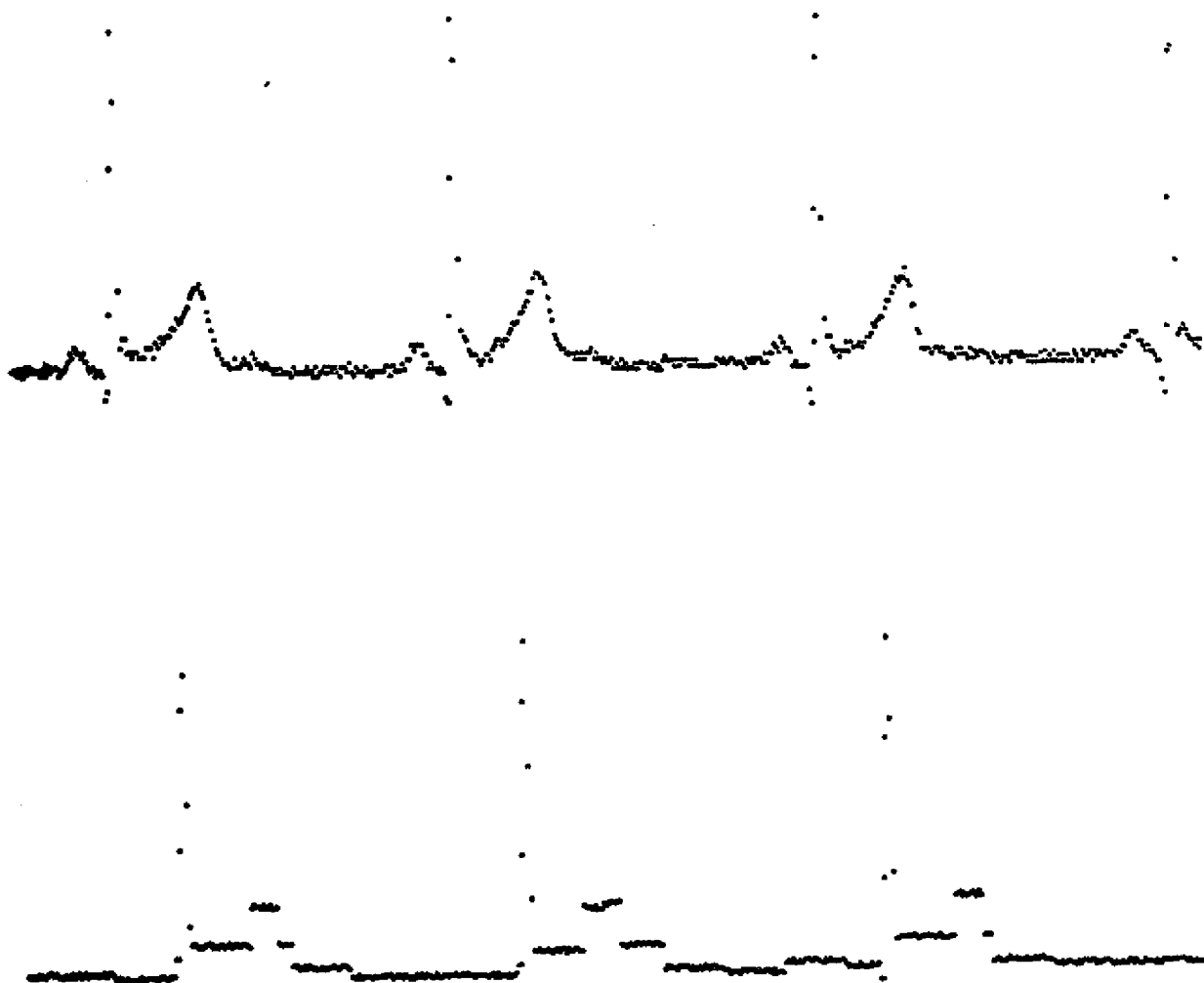
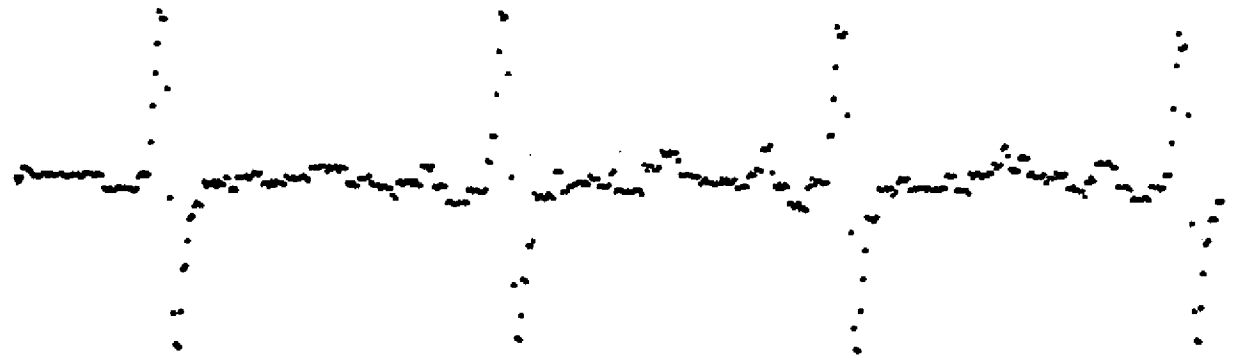
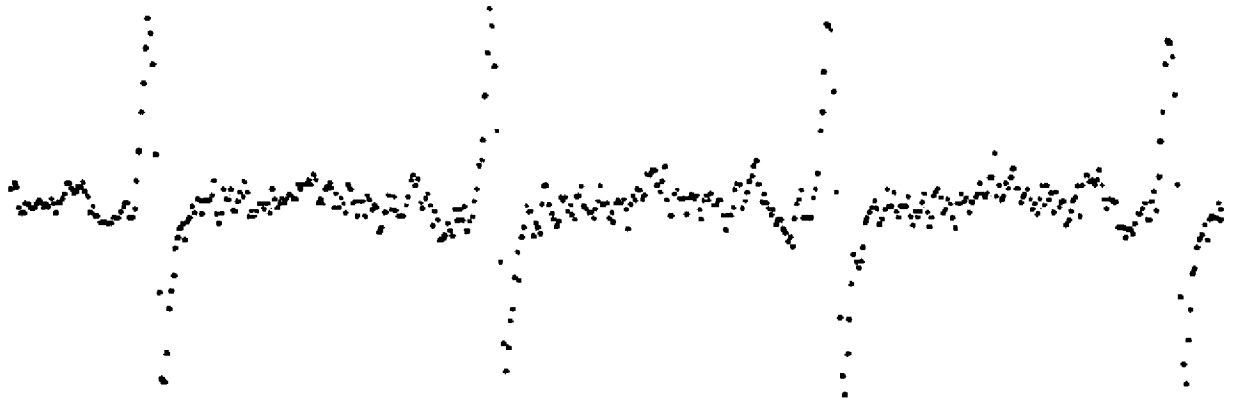


Figure 12: Interpolation of a noisier recording, default parameters. This tracing was done from data originally gathered by American Optical. The large amount of high frequency noise is inadequately represented by the sampled signal. The noise is reduced somewhat by interpolation, but there remain a large number of short segments in the baseline. These present a problem as far as degree of data reduction is concerned, and as far as recognizing legitimate steep slopes is concerned.



**ORIGINAL PAGE IS
OF POOR QUALITY**

Figure 13: Interpolation of a noisier recording, with aperture twice normal. This is the same waveform as figure 12, with the aperture at twice the coarseness. The baseline is now very reasonable, but note that many of the segments in the QRS are now two units long. This is not necessarily a problem, but must be taken into consideration by the detection algorithm.

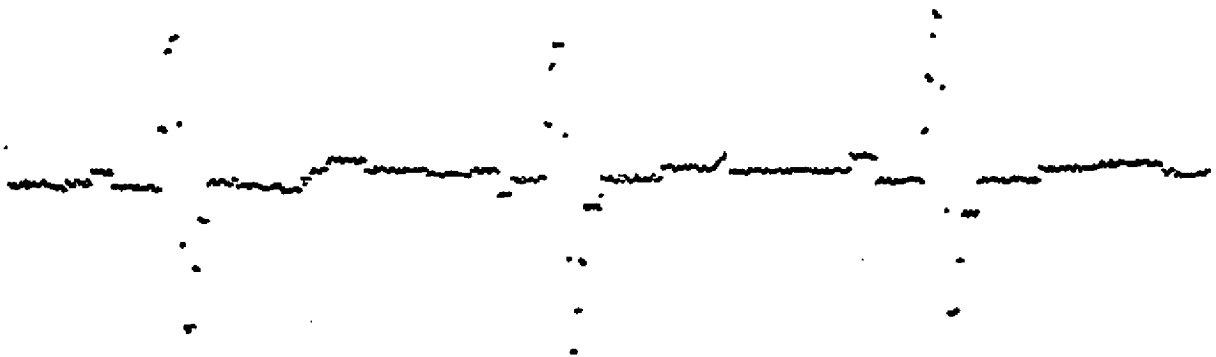
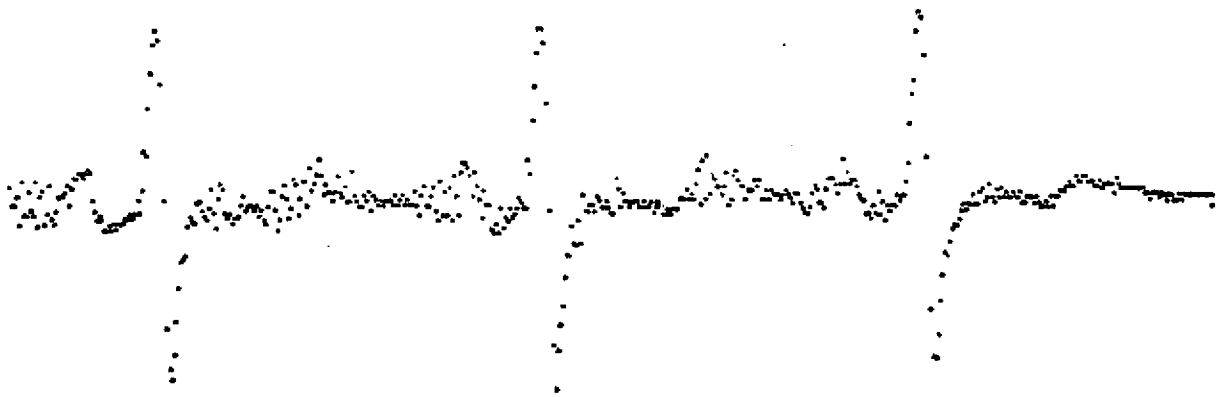
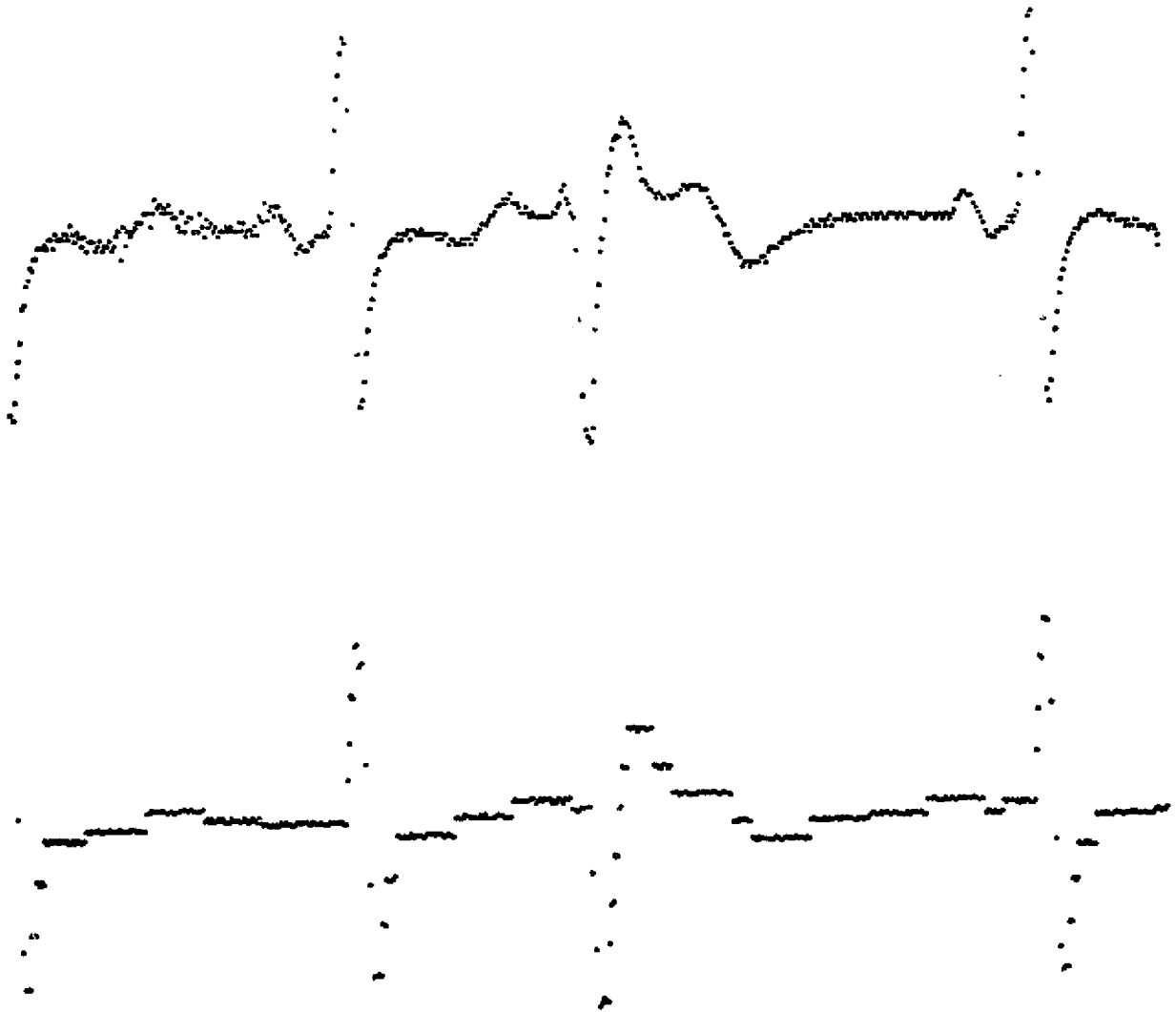


Figure 14: Interpolation of a PVC, with aperture twice normal. This display is from the same recording as figures 12 and 13. Note that both the PVC and the normals stand out well from the filtered baseline, but that again care must be taken to ensure that segments within the QRS of lengths greater than 1 unit are properly interpreted.



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 15: Interpolation of a second PVC, default parameters. This waveform is from a tape recorded at Peter Bent Brigham Hospital. The very clean signal depicts well the features of the algorithm. The negative portion of the PVC is flattened due to saturation of the analog/digital converter.

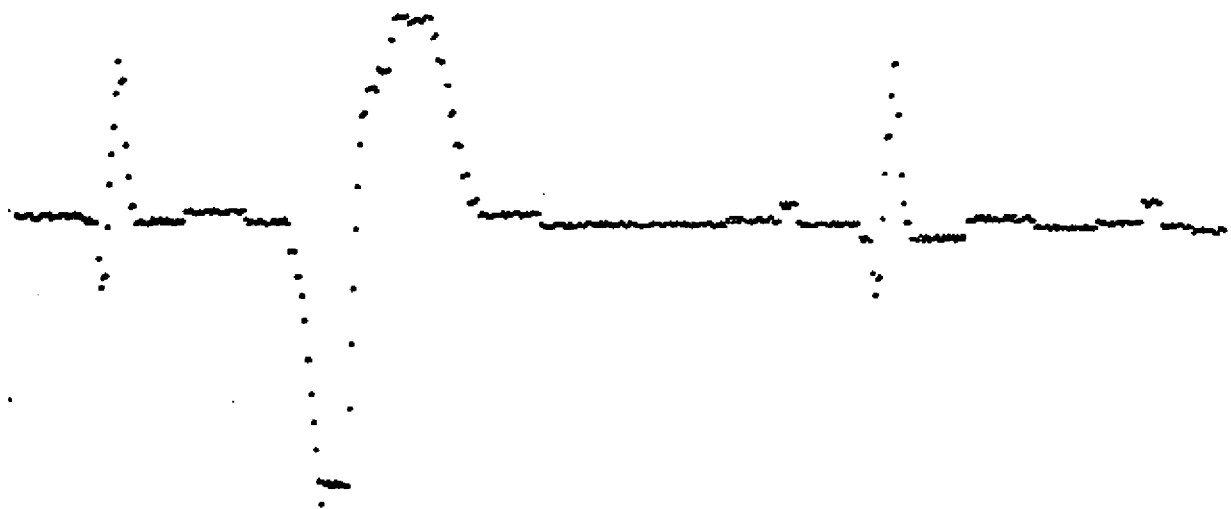
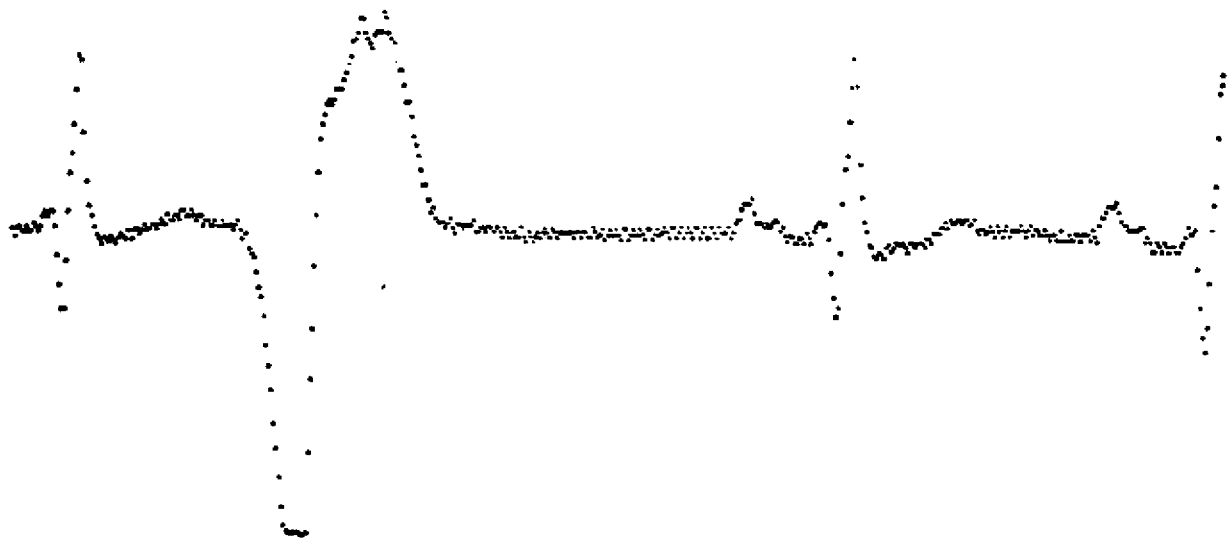
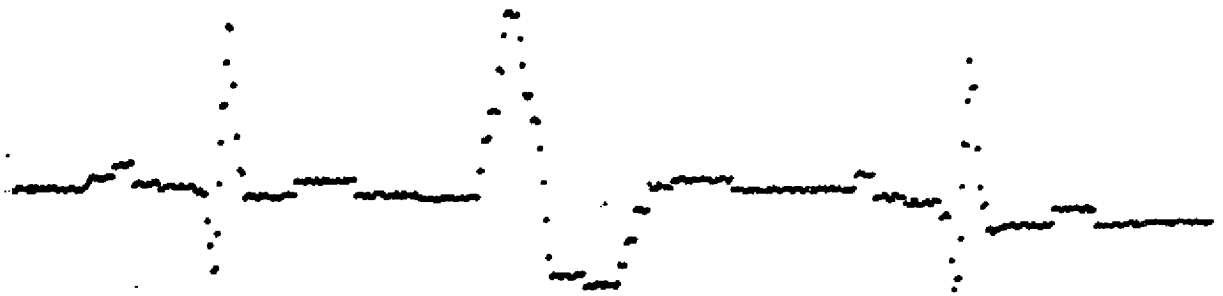
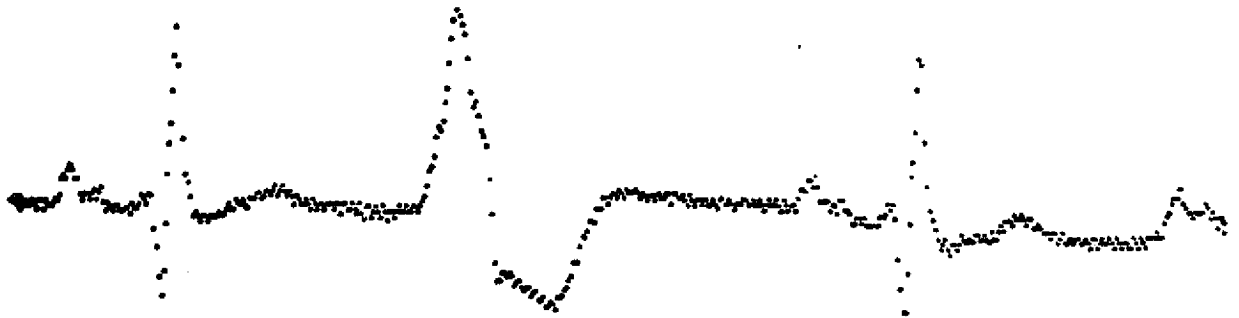


Figure 16: Interpolation of a third PVC, default parameters. This is a different PVC from the same PBBH tape. Again, the necessary features are all well-defined, but the detection algorithm must be carefully designed to abstract the proper conclusions about onset and end of the waveforms.



ORIGINAL PAGE IS
OF POOR QUALITY

Currently, the default value of this aperture is about 6% of a normal QRS peak-to-peak amplitude. As can be seen, the QRS is preserved nearly exactly, all significant waveforms are still easily recognizable, and the number of line segments is approximately a factor of 10 lower than the number of points input. The sampled signal and the output of the ZOI are displayed on the display scope. The figures included are all drawn directly on the X-Y plotter, using the option described in the section on how to use the routine, and are the same pictures that appear on the display scope.

Detection

This part of the program attempts to find all normal and abnormal QRS complexes, and reject everything else. Its basic premise is that QRS's are comprised primarily of large slopes, most large slopes are in the QRS, and the QRS is larger than the rest of the ECG. Initially, the program waits until a long flat section has passed, to ensure that its detection begins between complexes. When a segment of critical slope is found, it is labelled as a possible QRS onset. As long as the signal remains steep, it is considered part of a possible QRS. A certain length of time after onset, the peak-to-peak amplitude so far observed is compared with a fraction of the running average of the total peak-to-peak amplitudes. If it is smaller, the event is rejected as being too small for a QRS, and a new search is begun. When a flat stretch of specified length is detected, it is

considered a possible QRS end. If it is further from the average baseline than a certain fraction of the average peak-to-peak amplitude, that flat stretch is considered to be an elevated part of a PVC, and the search for a QRS end continues. Otherwise, it is decided that the full event has been detected. If the event is narrower than a specified cutoff, it is termed a noise spike, otherwise it is a valid QRS complex. Also, all QRS's are automatically terminated after a certain length of time (such as 200 msec.) if they have not already been by a flat segment. The flow chart in figure 17 illustrates this algorithm. After the end of the QRS, a certain length of time (150 ms. default) is ignored so that no large, steep T-waves will be mistaken for QRS complexes.

Reasonable baseline shifts, small amplitude waves and noise, and narrow noise are all successfully rejected by this algorithm. One critical parameter must be adjusted to differentiate between thin QRS's and noise spikes. On recordings with no more noise than can normally be expected in a clinical environment, this module does an excellent job of finding all QRS's and ignoring the other events. There is a large room for modification and improvement to upgrade the detection under more severe environments. The onset and end are not always found precisely on difficult samples, but this is more of a measurement problem, as explained in the next section. The critical slopes, the critical flat sections, the times where different parameters are observed,

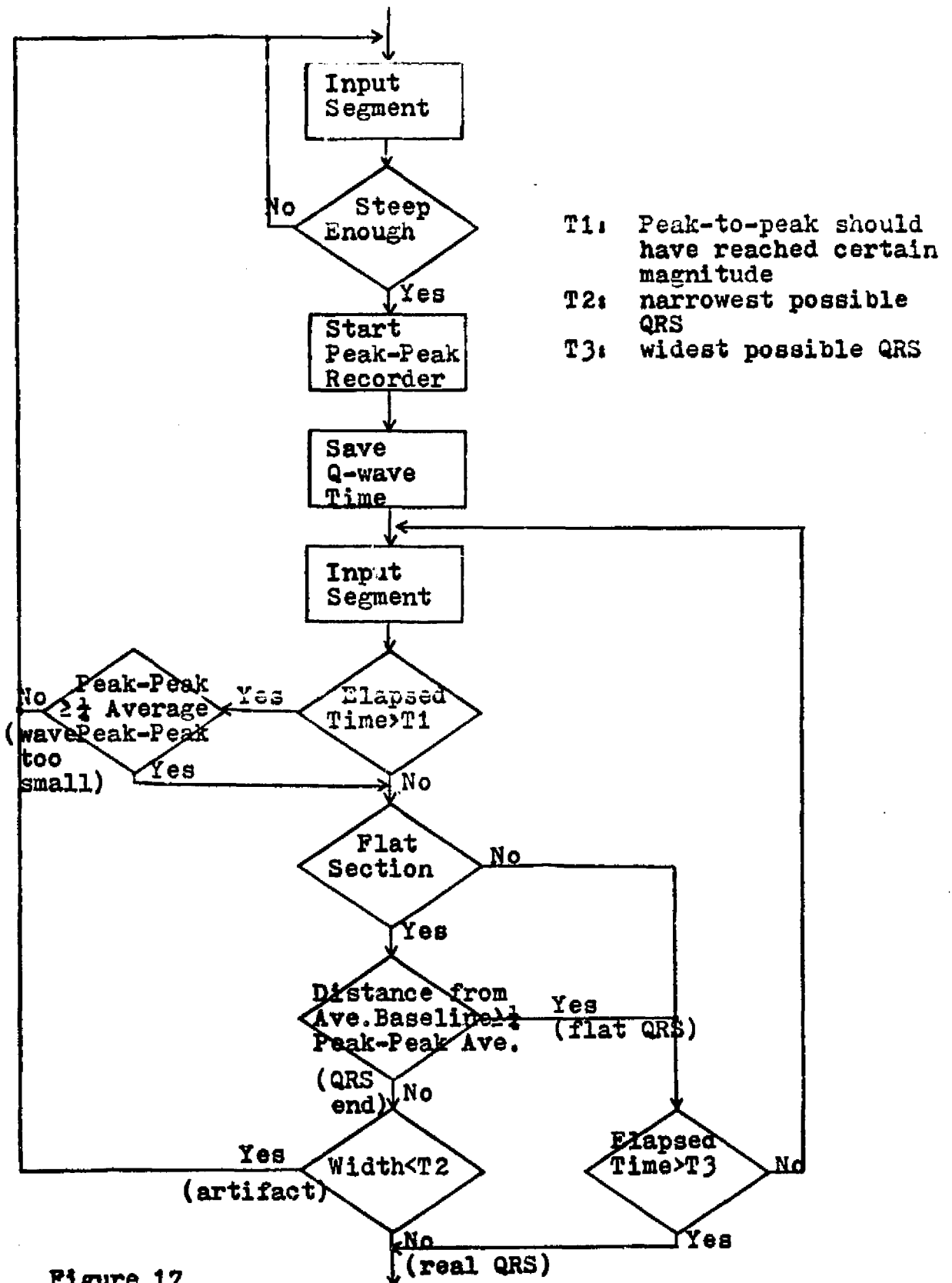


Figure 17

Flowchart of detection algorithm

and the amplitude cutoffs for event decision are all single parameters that can be adjusted with experience and for data that is unusual. The enclosed sample strip charts illustrate some of the successes and failures of this routine.

Measurement

For each QRS that is detected, certain parameters have to be calculated. Their accuracy is heavily dependent on the success of the detection module. Width and magnitude of the QRS, QQ interval between QRS's, and baseline amplitude are measured. In order to make time calculations, an auxiliary variable called SegmentClock has been created. As each line segment is read from the circular buffer its length is added to SegClk, to uniquely (modulo $2^{16} \times 5 \text{ msec.} = 5 \text{ min.}$ at real-time) specify its temporal location. The width is simply elapsed time between what was detected as onset and end of the QRS, QQ interval is between consecutive onsets, and magnitude is determined by a running window which expands as each new maximum or minimum within the QRS is found. Baseline for a particular complex is defined as the amplitude of the last flat segment prior to the complex. If, as in a very large PVC, the input signal saturates the analog/digital converter, the magnitude is calculated as if the signal had barely touched the extremum. The measurement routine is consistent, as observed on several identical analyses of the same data. The maximum width measured is determined by the detection routine, now set at 200 msec. The program halts if it finds a QQ interval less than 240 msec. or greater

than 10 seconds. The variables that affect measurement are those mentioned above in detection, primarily for making accurate decisions on where QRS onset and end are. The main cases of inaccuracy are steep baseline shifts and noise which camouflages onset and end. Coarsening the interpolation process does help with low amplitude, high frequency noise. Shortening the flat stretch indicating an ST segment allows accurate detection of end where it would otherwise be missed in noise, but it reciprocally introduces a hazard of terminating too early. Permitting a flat segment far from the baseline to be called a QRS end instead of a flat segment of a PVC filters out baseline shift better but increases the probability of mis-measuring PVC parameters.

Decision

Presently, only binary decisions are made on PVC versus normal QRS. All data is available to classify PVC's into families having similar morphology if that is so desired. To find abnormal beats, the measured parameters are compared to running averages determined from the normal beats of that patient. The relative weighting of the averages is adjustable, presently the most recent parameter is given $1/3$ or $1/7$ the influence of the previous average. A PVC is found when the current width is significantly larger than the running average, or the QQ interval is significantly shorter. The magnitude is not considered in this algorithm. The actual cutoffs are easily adjustable, with defaults at 175% of average width and $15/16$ of average QQ. These are

important parameters to consider changing, as more data yields a better foundation and as a function of the individual patient's ECG. For example, a person whose normal heart rate varies considerably with breathing would want a less sensitive QQ cutoff. It is essential that running averages be used as the standard, since normals vary widely between patients and over time in the same patient. The first several complexes in each new tape are automatically considered normal, in order that their parameters may be incorporated into an initial running average for that tape.

A more complex decision tree could be easily implemented. Magnitude of the QRS and information on compensatory pause are accessible, and a decision that considered the dependency of variables would also be feasible. This module is the main area where tradeoff between false negatives and false positives is determined. Tighter bounds will let fewer abnormals by, and also classify more normals as abnormals. The current fairly loose bound on the width is due to the fact that measurements of width are not highly precise. So far, the decision algorithm has proved accurate, with blame for error resting primarily on detection and measurement.

Output

The output consists of a single channel where the amplitude, width, and QQ interval appear coded as three pulses whose height is proportional to the value. Another pulse, opposite in polarity to the first three, appears when there is a PVC. The distance between sets of pulses is

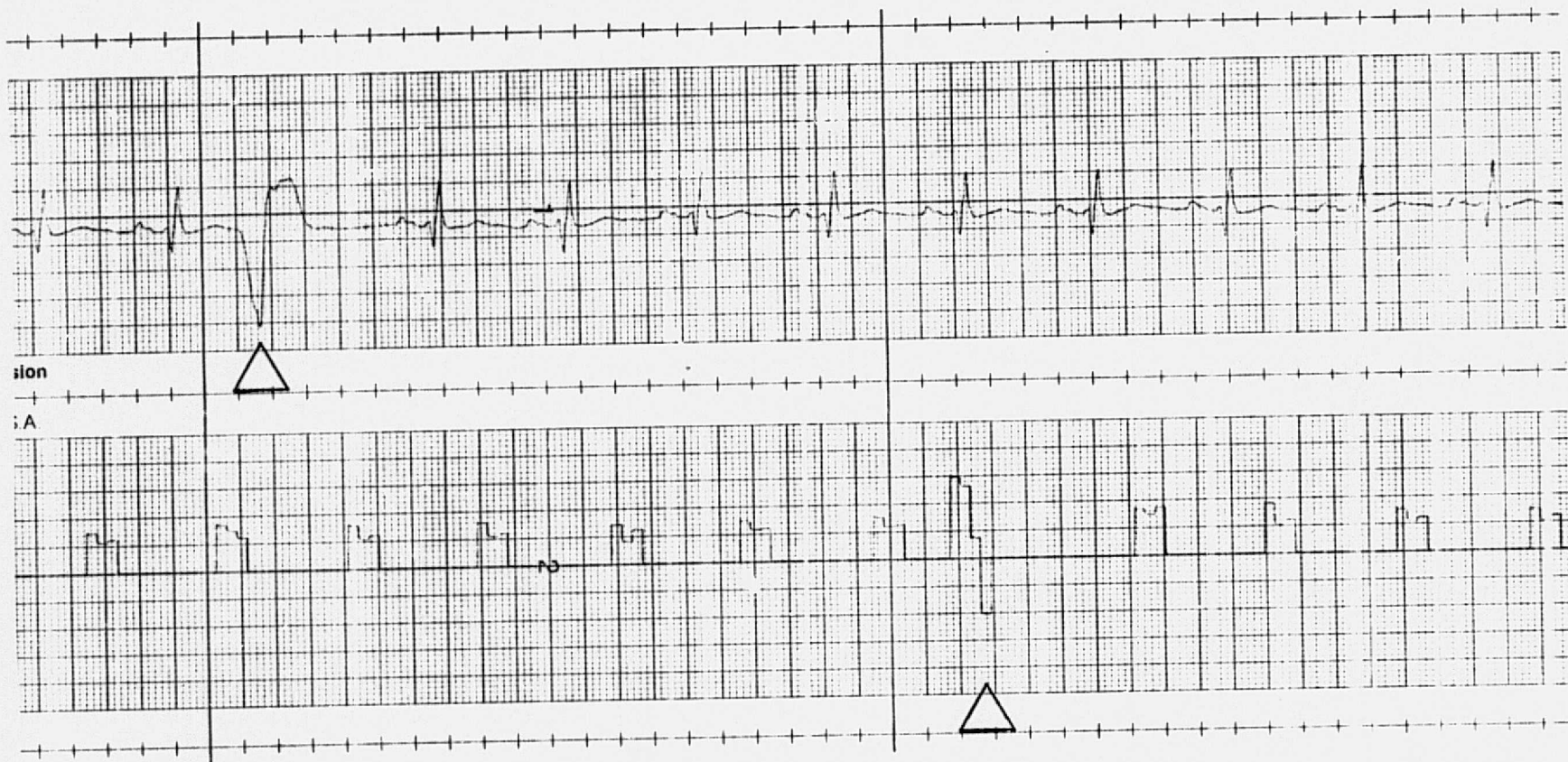
equal to the QW interval minus the total width of the pulses so that the output remains a constant time delay behind the input ECG. This delay is adjustable. The output parameters are stored in a circular buffer to allow for variable delay and a certain amount of variation in speed of analysis. It is expected that this data channel would either be recorded on a separate track of the ECG tape, or the ECG and the data would be recorded on 2 channels of an output magnetic tape. If all that is desired as an output is a set of internally generated statistics on PVC's, there is no need to record the data channel. Otherwise, it is intended that the coded channel would be usable for further analysis and to enable the PVC's to be accessed very easily, such as for obtaining strip chart recordings of only the abnormal beats. More ideas on this are presented in the paragraph below on future options. Figures 18 through 27 illustrate many features and problems of the system as recorded from clinical data. Further explanations are included with the recordings.

How to Use

The use of the total program is very simple. First, the sampling rate must be changed if it is to be other than real-time (or whatever other permanent rate has been installed). If any other parameters are to be changed from default values, that should also be done at that time. The four sense switches should be off, for the display mode. The tape recorder is attached to the Analog Input, and run at the appropriate speed. A dual trace now appears on the display

Figures 18 through 27 were produced on a strip chart recorder with tape speed 25 mm/second unless noted otherwise, and sensitivity 100 mv/division on both channels in all cases. The ECG waveform is the raw signal from the tape recorder, while the lower channel is the output provided by the computer via the Analog Output. The height of the three pulses are proportional, in order, to the peak-to-peak amplitude of the QRS complex, the width of the QRS, and the QQ interval. A fourth pulse, of a constant negative value, is present in the case that a QRS is declared to be a PVC. The data channel is delayed behind the QRS by an adjustable amount, in these cases about 4 seconds (the full-width vertical red lines are 4 seconds apart).

Figure 18: A detected PVC. The large PVC at the left has been detected, its flag is to right of center. Note the similarity of the coded parameters for the other QRS's, as contrasted to the large amplitude and width and short QQ interval indicated for the PVC. The magnitude is not considered in the current decision algorithm, but either the width or the interval would have classified this as a PVC. The following set of parameters indicates a large QQ interval, the compensatory pause. The delay between PVC and flag is the appropriate 4 seconds. This was recorded from a PBBH tape.



ion

A

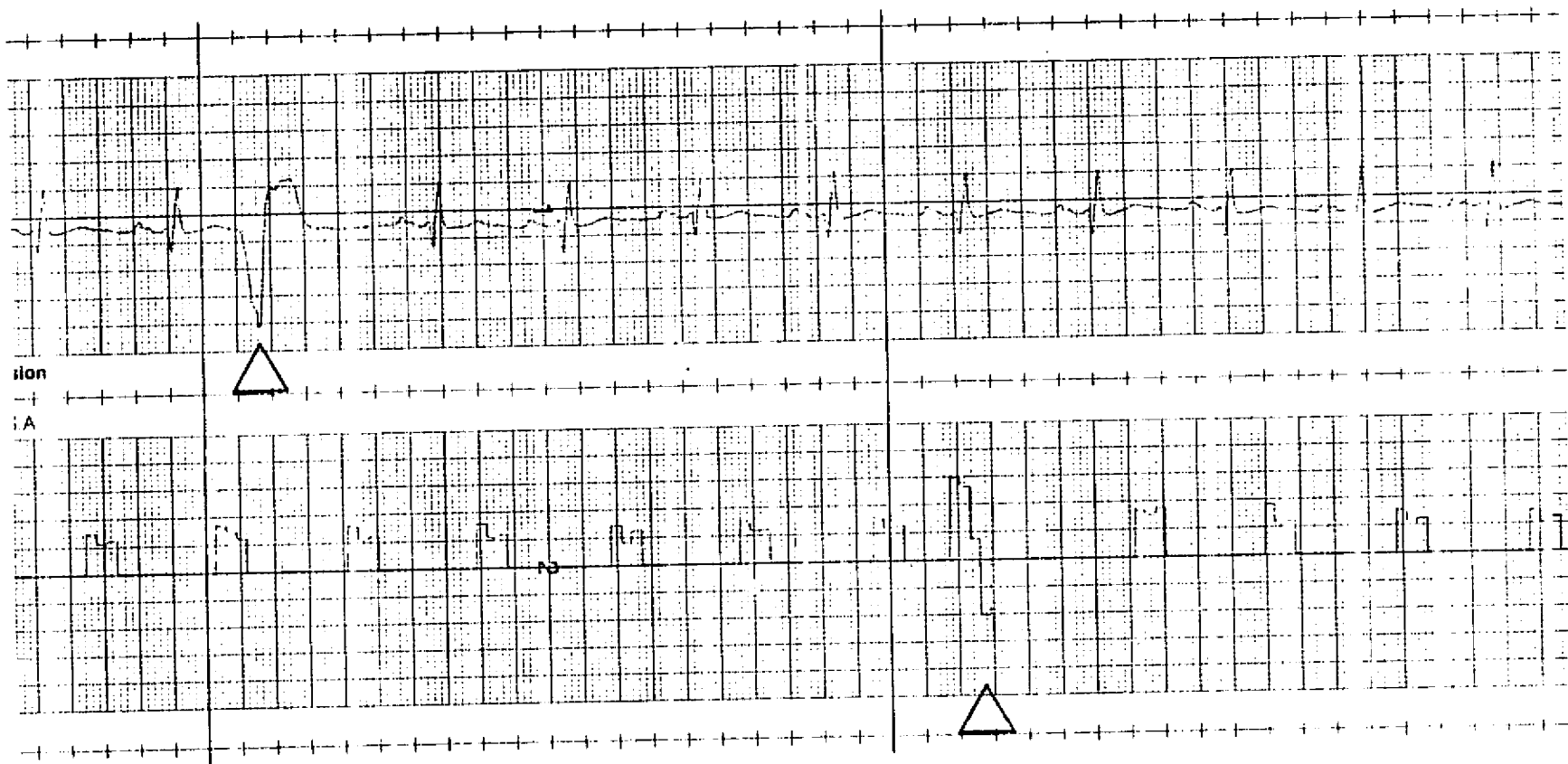
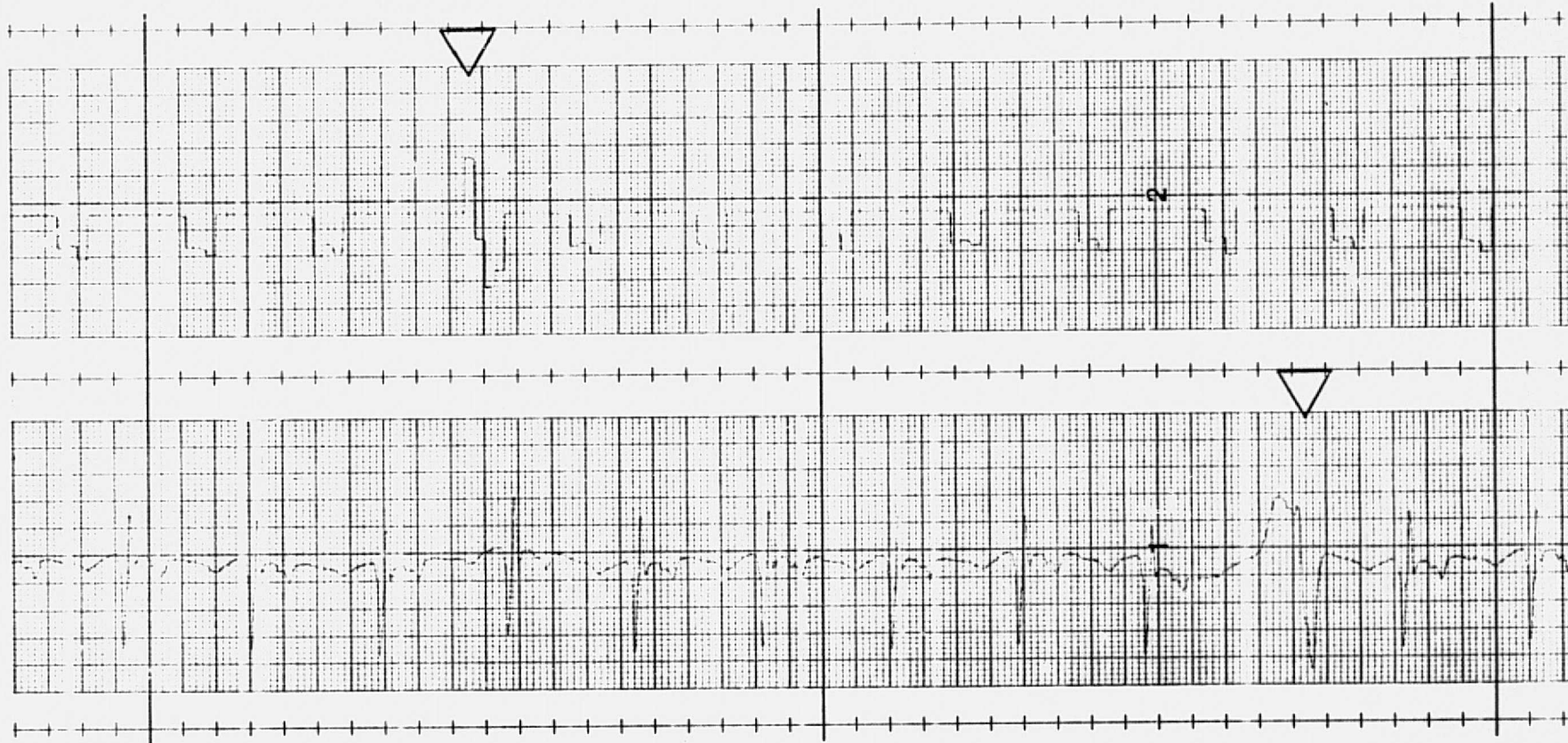


Figure 19: Another detected PVC. As in figure 18, the PVC was recorded as being large in magnitude and width and short in QQ interval. The steep P-waves posed no problems. This is from the same PBBH tape, and the QQ cutoff was set at $7/8$ of average.



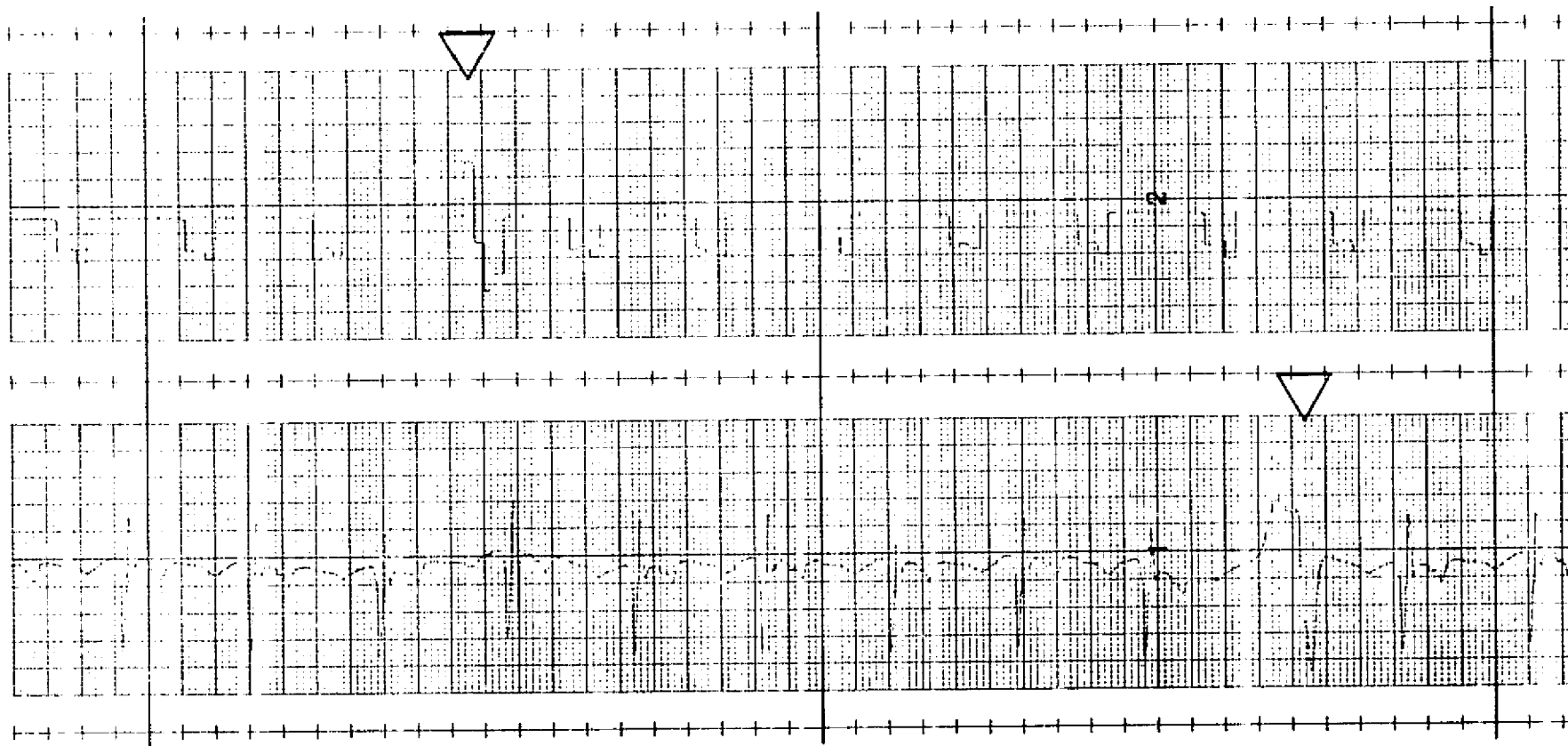
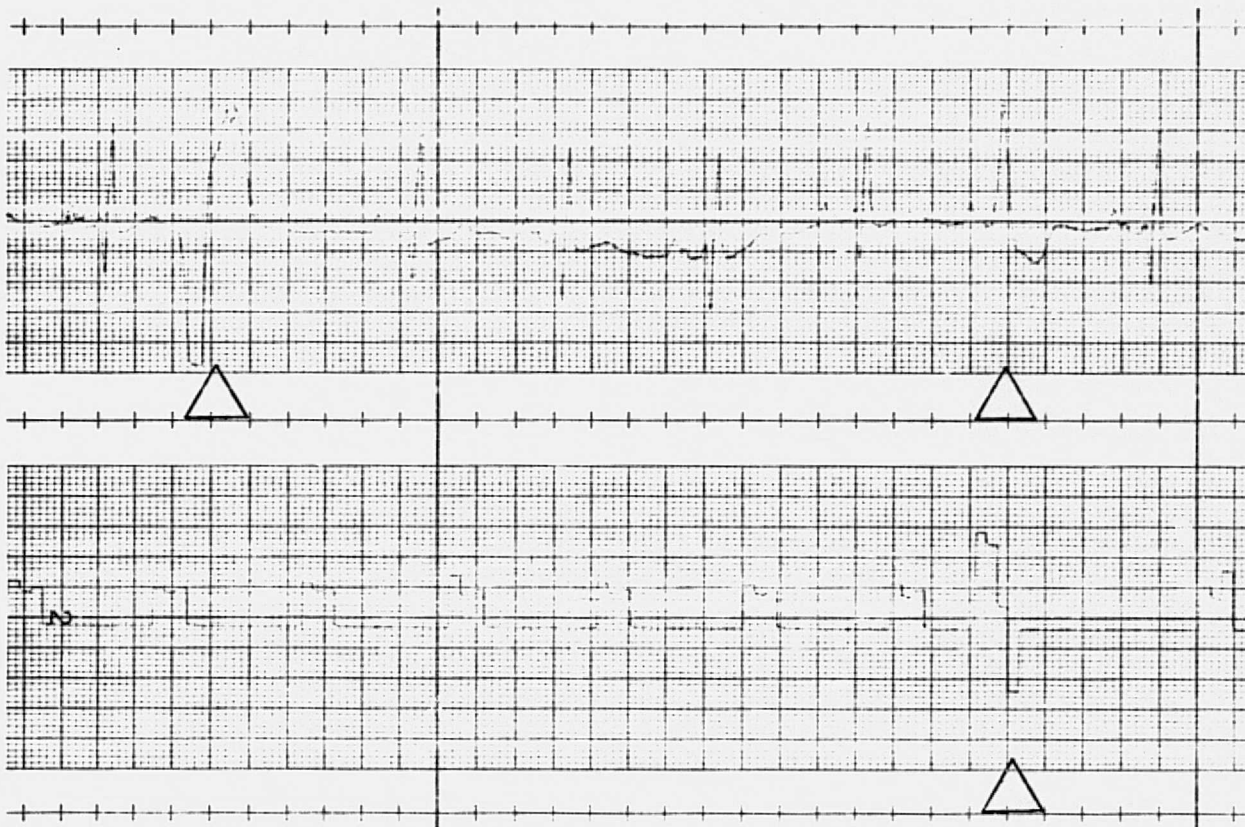


Figure 20: Two more detected PVC's. These two strips form a continuous segment of tape. At the beginning of each strip is a PVC, plus an abnormal non-PVC near the end of the first strip. The two PVC's were flagged as such, the other was not. Note the unsteady baseline. The abnormal non-PVC passed because it was not sufficiently premature, and its measured width was about normal. Again, the PBBH tape and QQ cutoff of $7/8$ average were used.



BRUSH ACCUCHART

Gould Inc., Instrument Systems Div

Cleveland, Ohio

Printed in U



Figure 21: A longer recording. This is a continuous recording made at 5 mm/second to allow a visualization of a longer pattern of detection. The only 2 PVC's were detected easily, and there were no false positives. It can be seen that the measurements are reasonable, and do follow slight variations such as magnitude from beat to beat. This is from the PBBH tape.

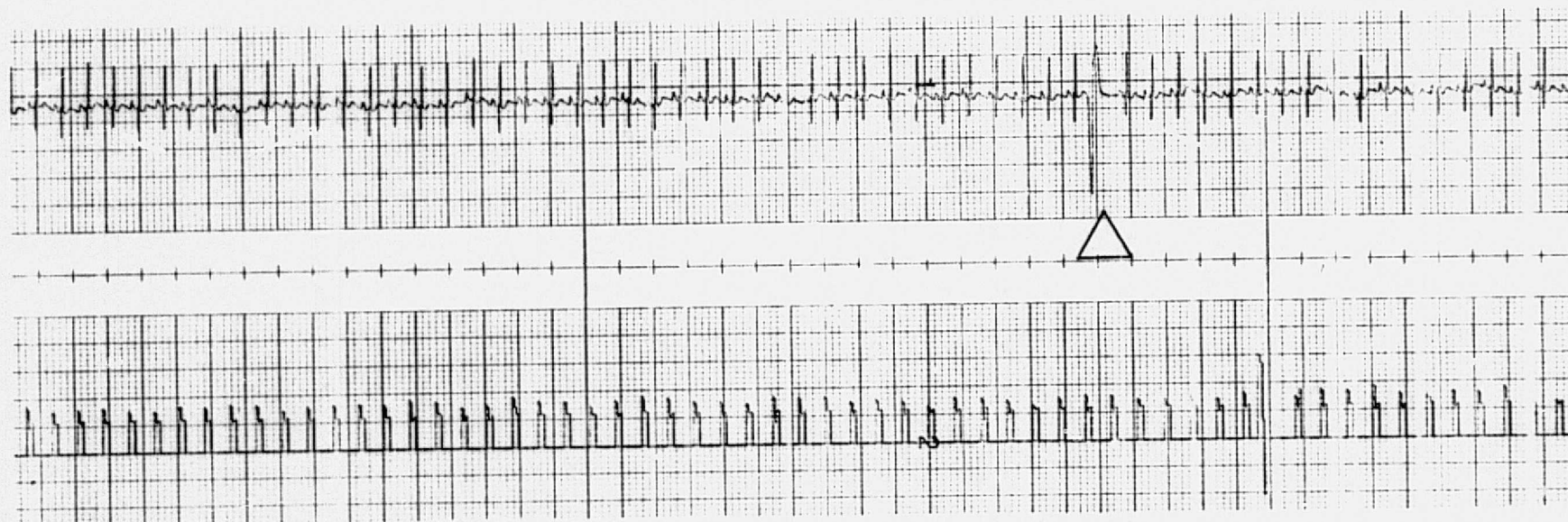
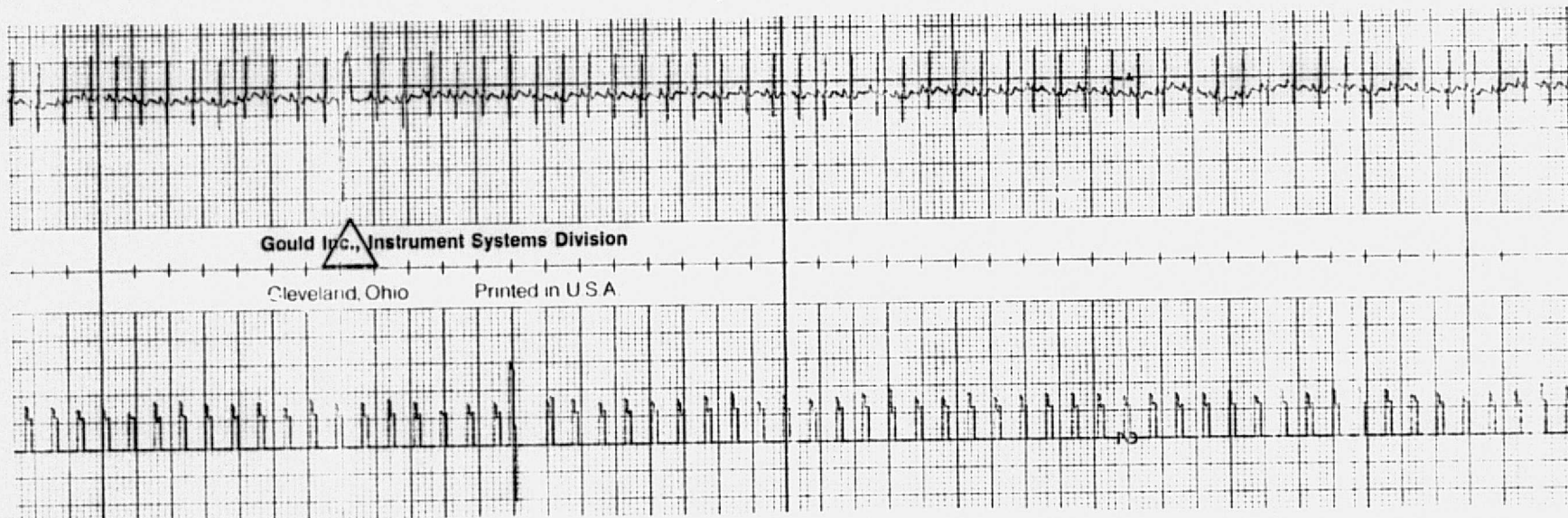
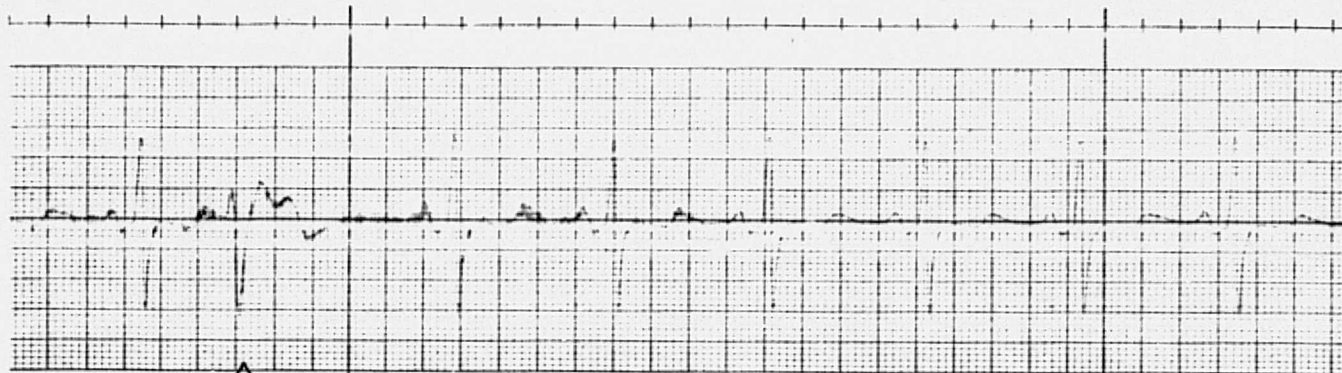


Figure 22: PVC's from another patient. This recording is from the American Optical tape. The QQ cutoff is $7/8$ of average interval, the interpolator aperture is twice normal, and the critical length to determine a QRS slope has been raised from 1 to 2 (in conjunction with the coarser interpolator). This continuous recording contains 2 PVC's plus fair amounts of high frequency noise. In this case the peak-to-peak amplitude of the PVC's are less than for the normals, the widths are of the same order, but the QQ intervals are well within the cutoff of $7/8$ of the running average. The noise has not significantly affected the measurements.



CHART



Gould Inc., Instrument Systems Division

Cleveland, Ohio

Printed in U.S.A.

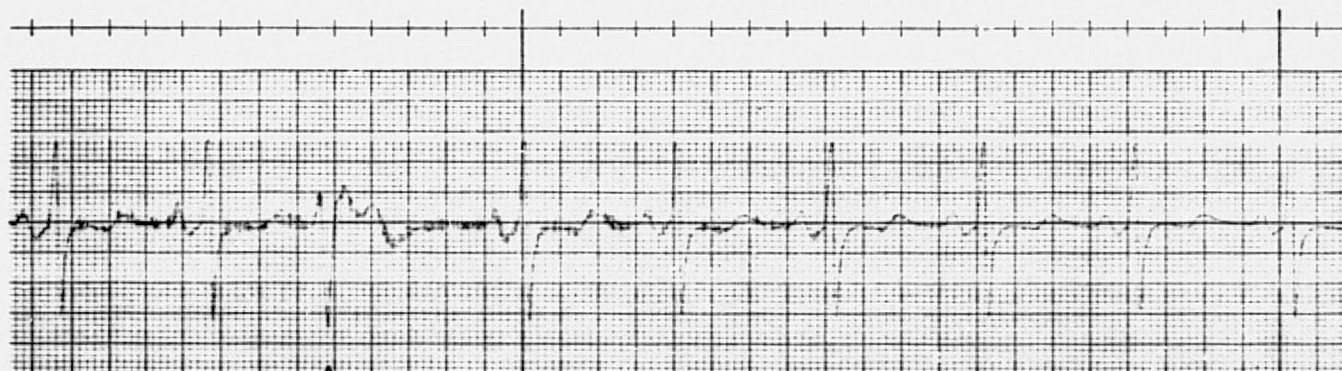
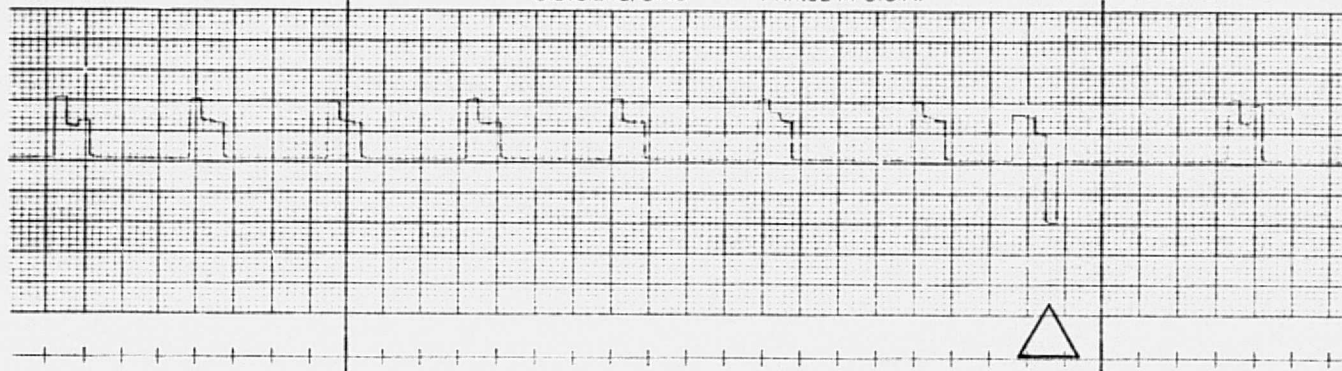
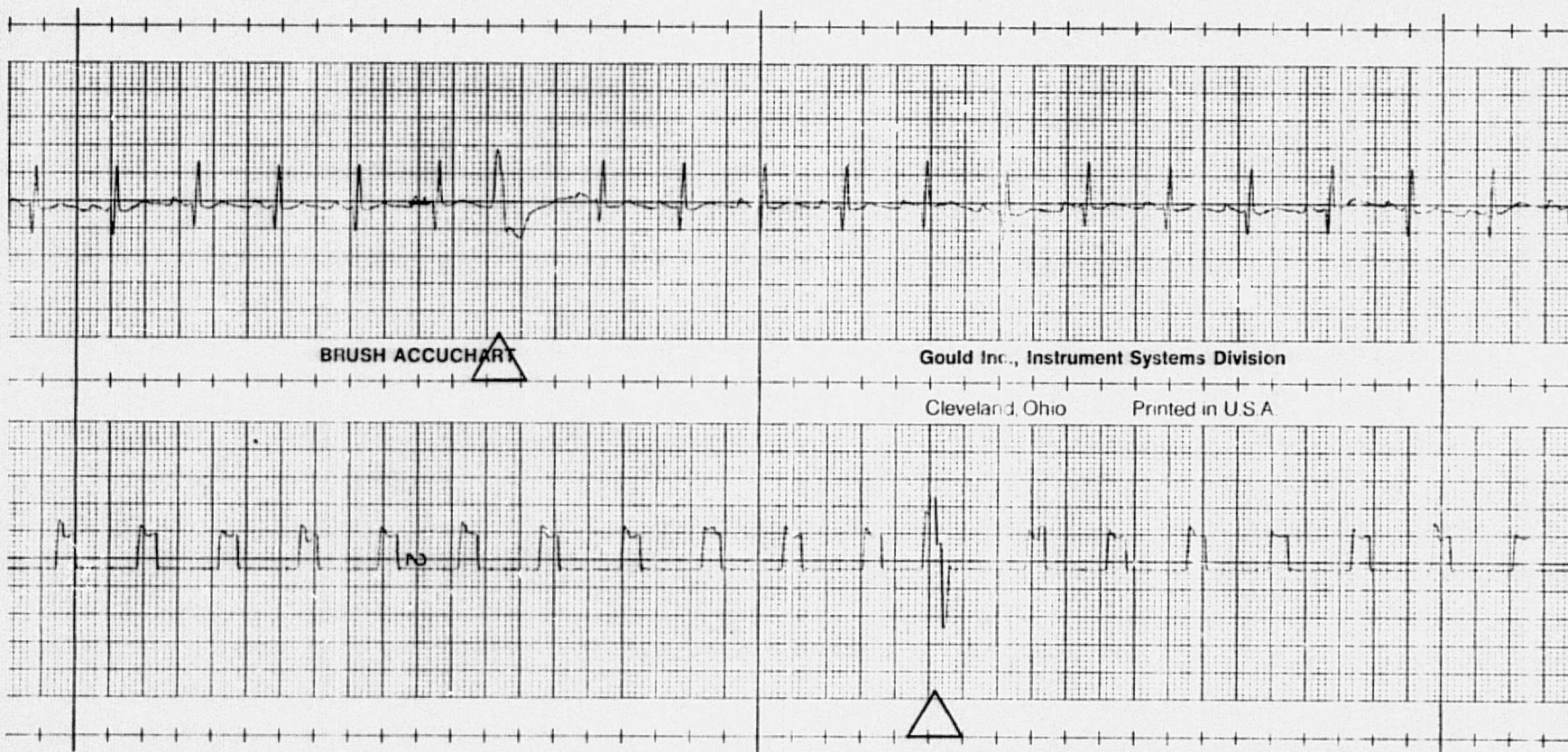


Figure 23: A direct recording at 8 times real-time.

For this sample, the PBBH tape was run at 8 times normal speed (and the sampling rate increased 8 times), and the strip chart was speeded up 5 times to 125 mm/second. The frequency response of the strip chart recorder is evident in the deterioration of the pulse shapes, but the algorithm and output work fine.



BRUSH ACCUCHART

Gould Inc., Instrument Systems Division

Cleveland, Ohio

Printed in U.S.A.

Figure 24: An indirect recording at 8 times real-time. In this example, the same tape was again played at 8 times the normal rate. But this time the output was recorded on a second magnetic tape (as would be expected in normal performances). This magnetic tape was played back at $\frac{1}{4}$ of its recording speed and the signal fed to the strip chart recorder, which was played at 25 mm/second. The analysis is the same as in figure 23, but the paper recording is enhanced.

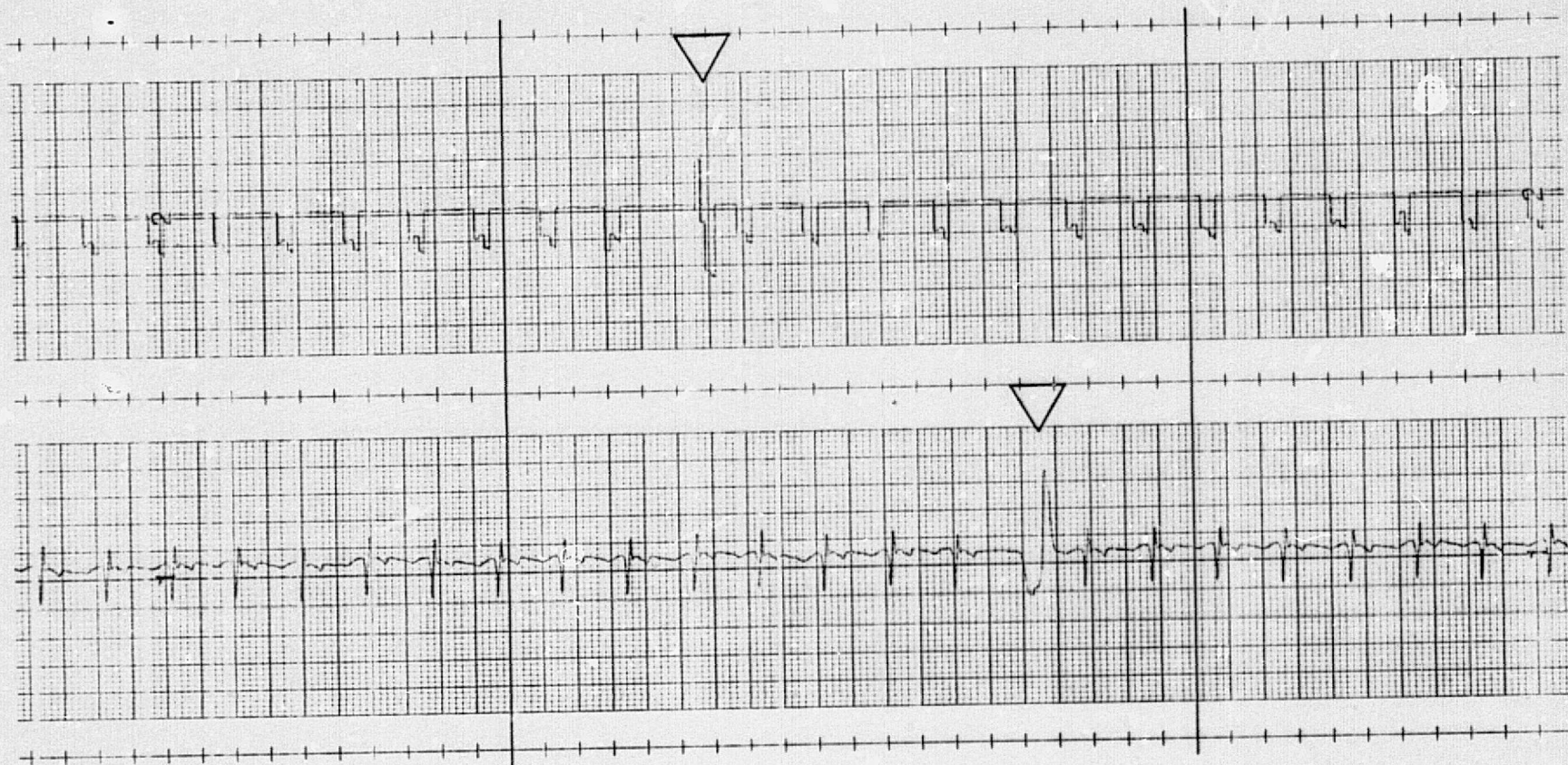


Figure 25: A false positive identification. This recording is from the American Optical tape. On the third QRS, there is a significant amount of steep noise at the S-wave. Due to this the end of the QRS was measured later, the complex was therefore incorrectly called wide, and the beat flagged as a PVC. Also, immediately prior to the fifth QRS is a negative, narrow noise spike. It can be seen that the program realized that it was a spike and reset in time to identify the Following QRS correctly. The aperture is twice normal, the length to determine QRS onset is 2 instead of 1, and the QQ cutoff is $7/8$ of an average interval.

0
L, 3

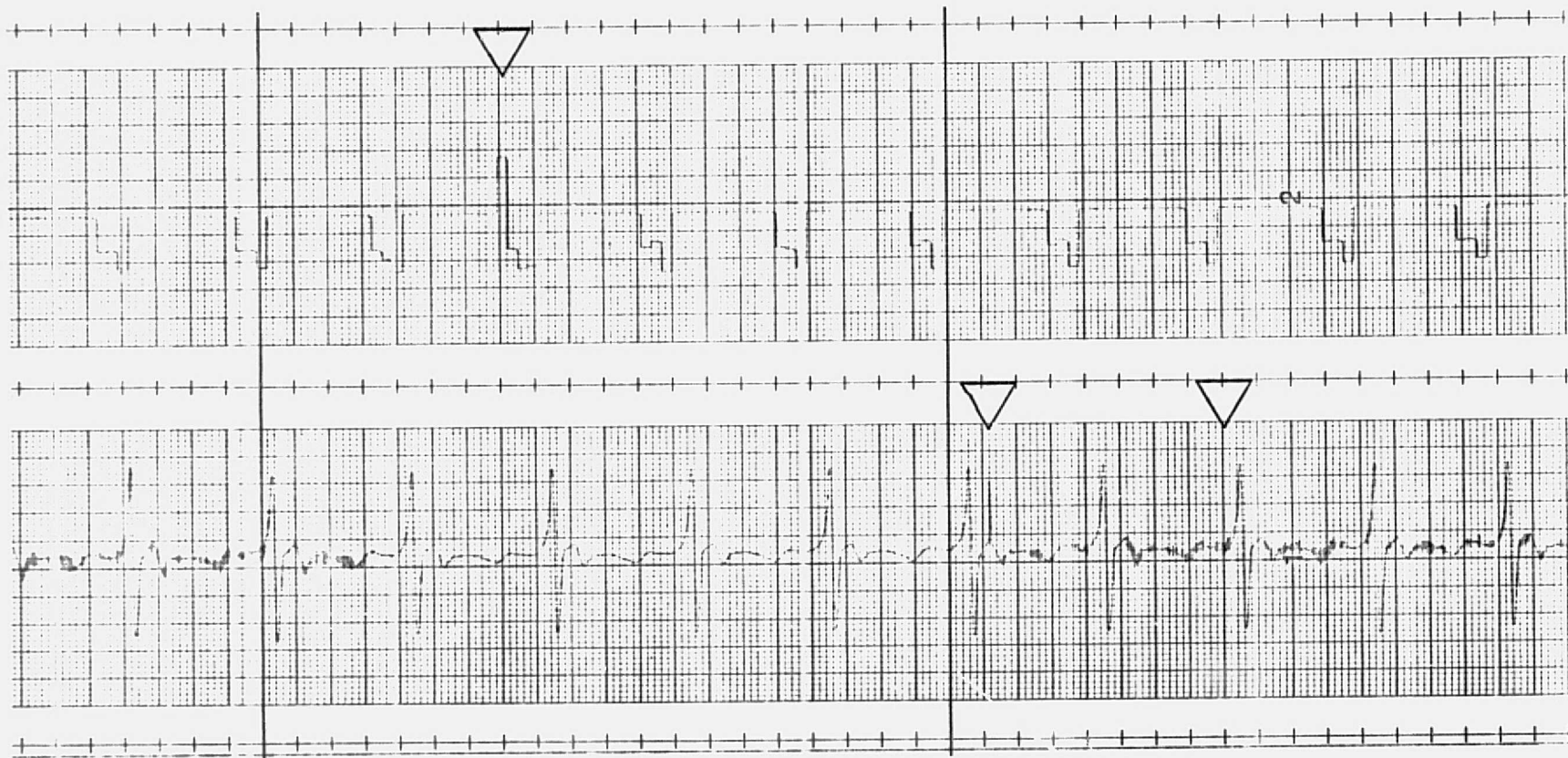


Figure 26: A second false positive identification.

In this case the large amplitude, steep noise was assumed to be a QRS complex. The amplitude was large enough, and the 2 largest spikes may have been observed as a single complex to yield an appropriate width. This pseudo-complex was considered premature and therefore the following real complex also had to be assumed premature. Both were flagged. The aperture was 1.5 times normal, the QQ cutoff was $7/8$ of an average interval, and the amplitude needed to declare an event a QRS complex was half of the average amplitude. This is from the American Optical tape.

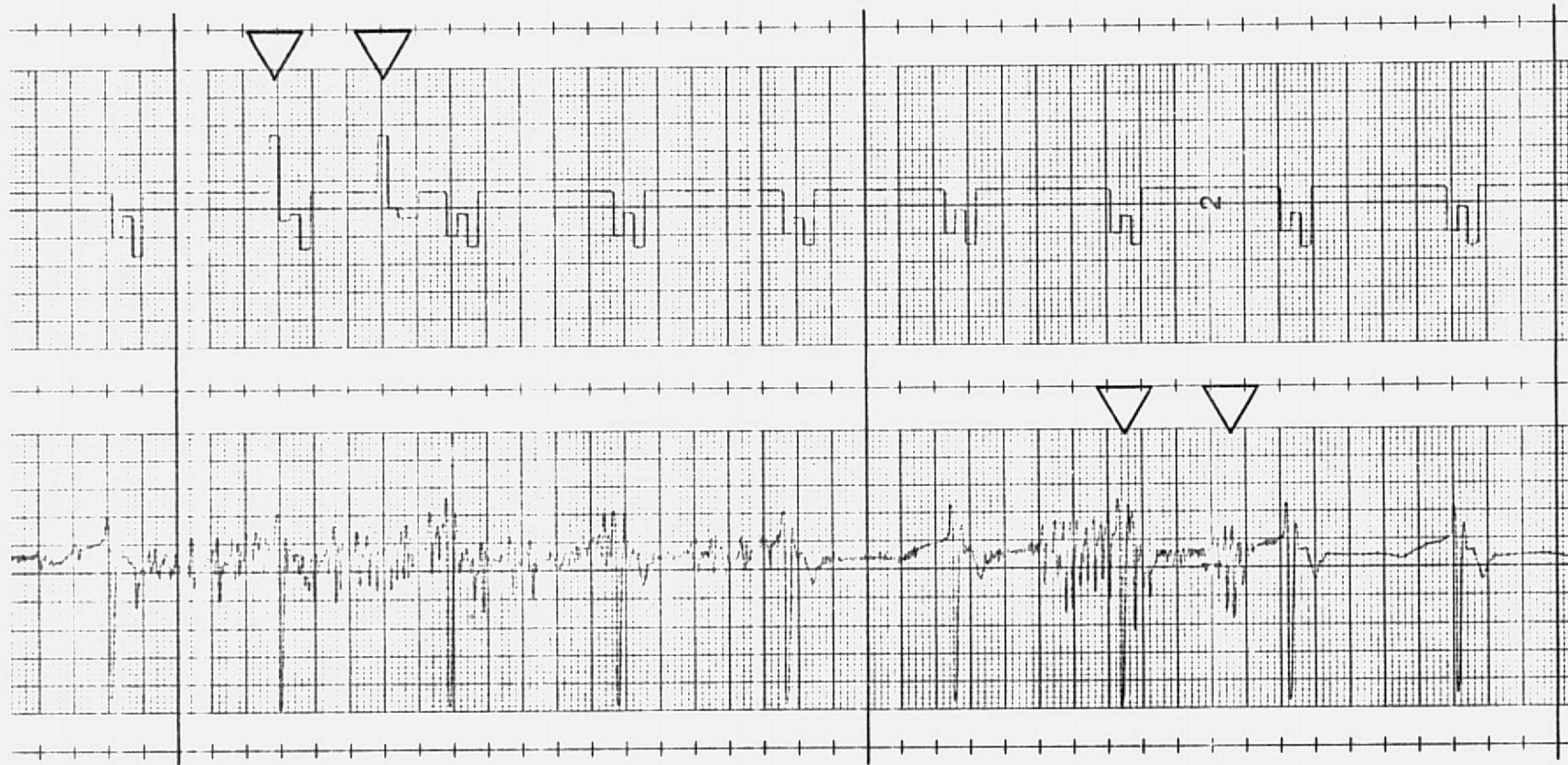
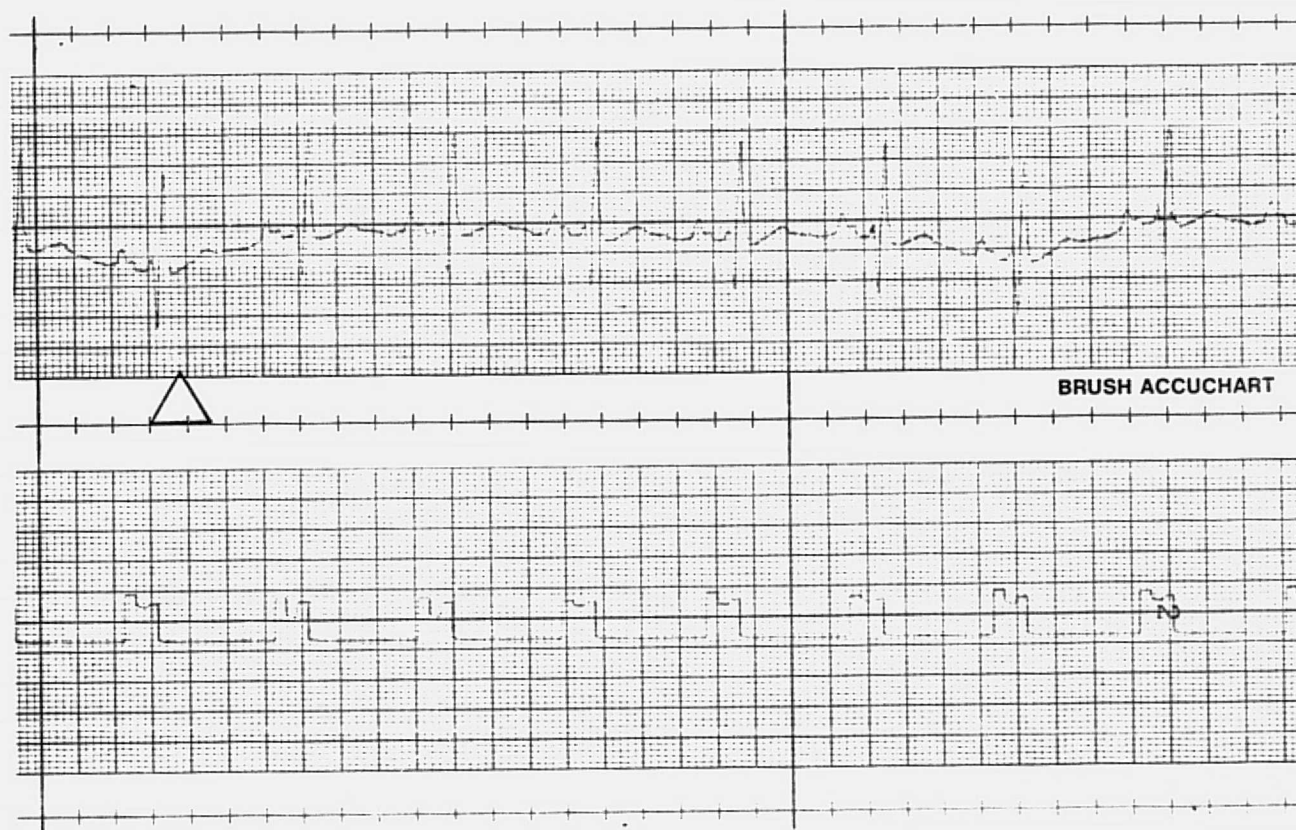
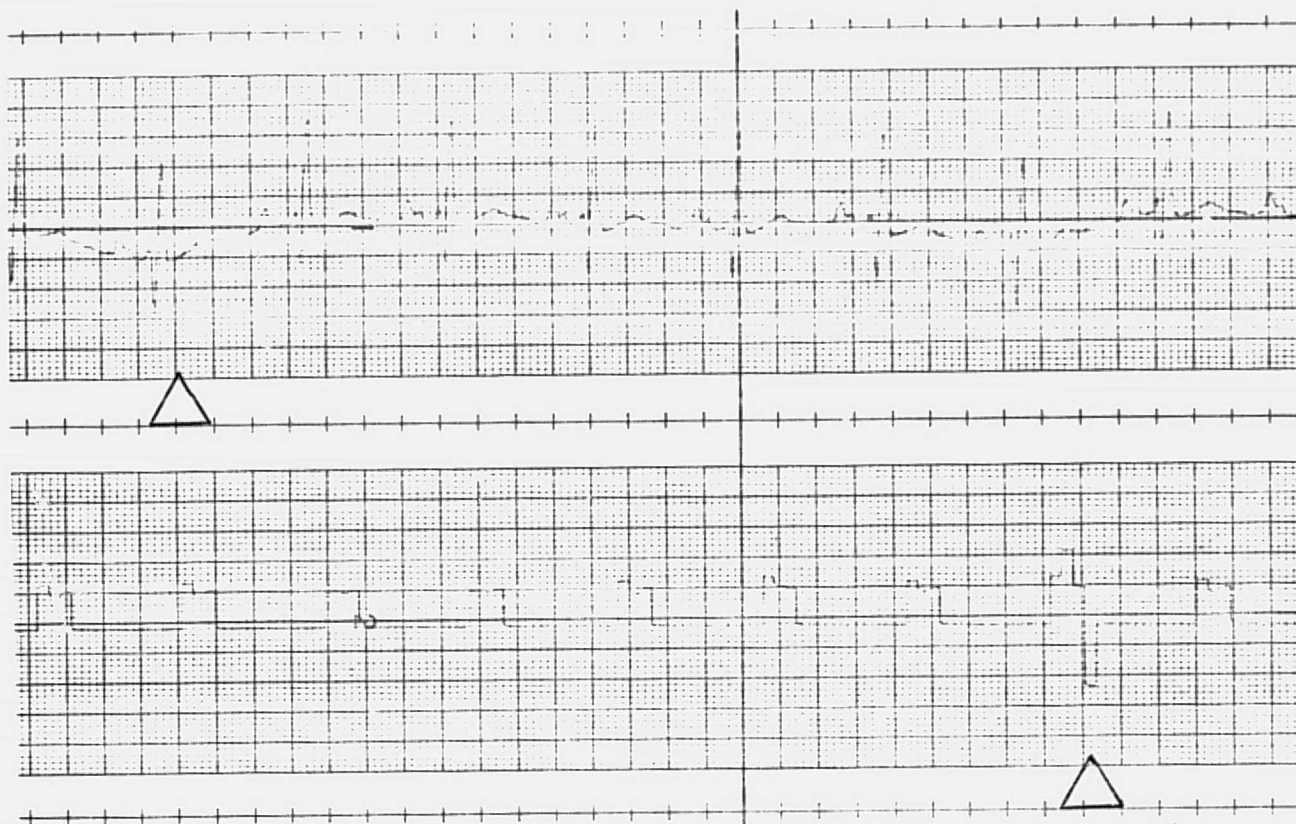


Figure 27: A third false positive identification, and its solution. The ECG sample for both strips is exactly the same segment from the PBBH tape. On the top strip the QQ cutoff is $7/8$ of an average interval. On the lower strip this is also true, and in addition the flat length needed to assume an ST segment has occurred is 30 msec. instead of the default 50 msec. On the top strip, the large swing in the baseline was sufficient to confuse the identification of QRS end. Therefore the QRS was assumed wide and flagged as a PVC. This problem was circumvented by allowing the QRS to terminate on a shorter flat segment, as can be seen in the lower tracing.



scope. On top is the sampled ECG, and on the bottom is the output of the zero order interpolator. The screen is continuously refreshed so that the traces appear to move slowly to the left. Figure 28 contains a flowchart of the display process. If sense switch 0 is raised in either the display or evaluation mode, the process returns to DOS control. Raising sense switch 2 freezes the display, so that a detailed look or photographs can be made. Lowering the switch returns the moving display. While sense switch 2 is up, raising sense switch 3 will enable the X-Y plotter and then halt so that paper can be loaded, etc. When the continue switch on the computer is pressed, a sweep is made on the X-Y plotter of whatever had been frozen on the display scope. As long as switch 3 is up, the same plot will be repeated. Lowering switch 3 returns the program to frozen display mode and disables the X-Y plotter. The figures included earlier were done this way.

The primary purpose of displaying the signal as the computer sees it is to allow the gain and offset to be adjusted to values which maximize information. It is recommended that normal peak-to-peak magnitude be adjusted to occupy about half the screen. When this has been done, sense switch 1 is raised and the program begins evaluation and output. The display stops, as this is very time consuming. Once the program is running, it will do so until a QQ interval is less than 240 msec. or more than 10 seconds, until an I/O device does not respond properly, until the program is

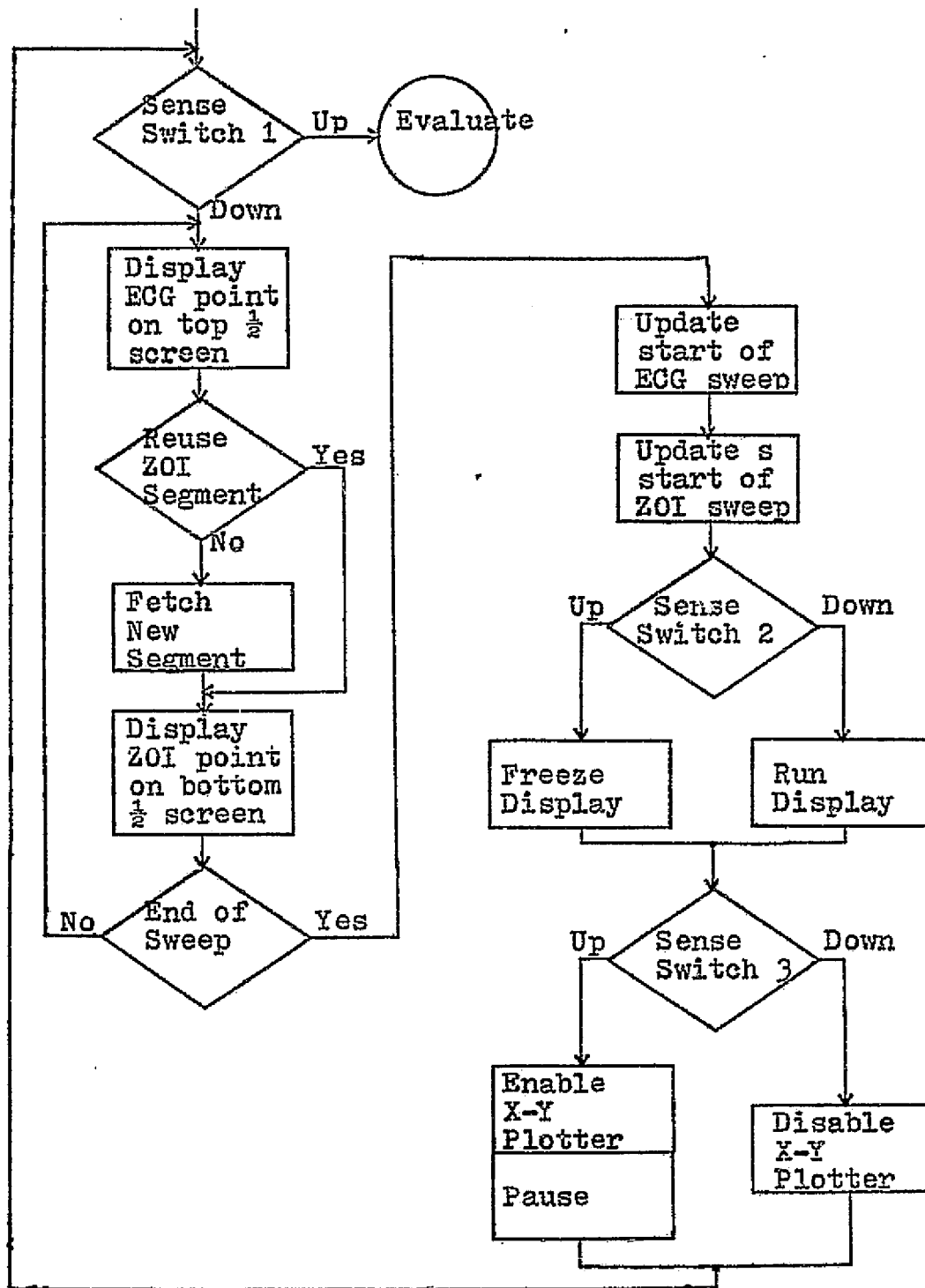


Figure 28

Flowchart of display algorithm

unable to handle the data rate as indicated by the data buffer being emptied, or until it is manually stopped. Restarting is easy, using the debugger, and is essential for a new tape because new running averages have to be established. If the program is run under debugger control, all features of that mode are available.

Theoretical Speeds, Running Times

At real-time, on the Nova computer, the ECG is sampled 200 sps, or every 5 msec. To handle each sample of input, about 50 instructions occupying 300 μ sec. are required. Therefore, about 60 msec. out of every second is needed to handle the input stream. This would indicate a maximum of 16 times real-time, for just inputting the signal, executing the ZOI, storing the data, and outputting the coded parameters and flags. To estimate the time needed for analysis of the wave, a heart rate of 80 beats per minute, or $4/3$ beat per second, is assumed. A generous estimate is to allocate 12 segments to the QRS complex and another 12 segments to represent the interval between complexes. Approximately 450 instructions are executed between each complex, 720 are used in analyzing the QRS, and about 250 are needed for handling bookkeeping after each complex. Therefore there are about 1420 instructions executed per complex and 1900 per second, still assuming real-time. This represents about 11 msec. out of each 1 second of data.

The total time required by the program is hence about 71 msec. out of each second at real-time. As can be readily seen, the time spent on analysis is a small fraction of the

ORIGINAL PAGE IS
OF POOR QUALITY

time spent on I/O handling. Therefore the estimate of 71 msec. is relatively insensitive to heart rate or number of PVC's, but rather primarily a function of the sampling rate. Using these numbers, a program should be able to run at 14 times real-time. If this is true, minor changes could be made in the instruction coding to increase the rate to 16 times real-time without changing the algorithm at all.

A large increase in speed would necessitate a decrease in sampling rate. For 120 sps, only $3/5$ of the amount of data is handled. The 14 times real-time would then go to about 21 times. Implementing any version on the Nova 800 is expected to yield an automatic increase in speed by a factor of 4 or 5. With this in mind, rates of 60 times real-time for the 200 sps model and 100 times for the 120 sps model are not unreasonable expectations. The 120 sps would of course lose some detail, but that is a rate which has successfully been used previously, as was referenced in an earlier section on background. Figures 10 and 11 were made with a sampling rate of 120 sps.

Analysis of Results

The four criteria of accuracy, speed, ease of use, and flexibility were presented as the requirements for a successful high-speed evaluation of ECG's. Accuracy has been established to the extent that it works on several tape-recordings which include PVC's. Several dozen sections of tape which were known to contain PVC's have been analyzed,

and all PVC's found. Very few non-PVC's have been labelled as such, and most of these false identifications can be eliminated with a more knowledgeable adjustment of parameters and criteria. The rate of false positives is generally below 1%, sometimes well below. This number is still higher than desired, but should be greatly reduced by using the full power of the program. A thorough statistical analysis of the accuracy under a variety of conditions is critical, and should be the next step in furthering this project. Based on experience to date, the accuracy of the program is good, particularly as regards having caught every PVC. The strip chart recordings included in this paper show a variety of PVC's that have been detected, illustrate a couple of situations that lead to false positive identification and point out how to eliminate some of these errors, and attempt to depict the overall accuracy of the algorithm.

The program has been run without any difficulty at 8 times real-time. Figures 23 and 24 illustrate this. Attempts to run at 16 times real-time start successfully but quickly halt when the demand for output exceeds the speed of analysis. This would seem to support the theoretical estimate of about 14 times real-time given above. This speed can be increased significantly as was described in the section on theoretical speeds. Although not outstanding, it seems reasonable that this operation can be run at 60 times real-time or more on the Nova 800, as

originally intended.

Ease of use is certainly inherent in this system. If no parameters are to be adjusted, the total operation consists of starting the tape recorder, adjusting the gain and offset as monitored on the display scope, and raising one switch to enter evaluation mode. Changing parameters is also very simple under debugger control.

The entire program is quite flexible in regard to modularity, changeability of parameters, and ease of adding new options. Entire blocks such as the detection and decision schemes may be replaced by others if so desired. As explained many times, individual parameters are easily changed, and should be to obtain better accuracy as better information is received and on patients with widely dissimilar ECG's. The present output provides a basis that can be used for a variety of output options, some of which are explained below under suggestions for the future.

SUGGESTIONS FOR THE FUTURE

This basic program can be improved in many ways, can be modified for similar purposes, and can have features added to increase its power. The first task should be to make a clinical, statistical evaluation on a large bulk of data. Presumably, the detection and decision blocks in particular have ample room for improvement. If that amount of increase of speed is desired, the idea of changing the sampling rate from 200 to 120 sps should also be tested.

This algorithm has been specifically programmed for PVC detection, but there is no reason that certain other arrhythmias could not be studied using a different set of parameters in the same basic framework. Even for PVC's, a completely different detection algorithm could be created. For instance, the initial detection could be based on amplitude instead of slope.

The output provides for the largest variety of options. This has been deliberately left open-ended, since each user will probably have different requirements. The program could be amended to do a statistical analysis of the input signal and provide a teletype output at the end of each tape. Or, the normal data channel output could be created and used as an input to a second stage hardware system that displayed all the PVC's on strip chart. Or, the program could store its output in digital memory without ever providing an analog signal. This idea could be very flexible

in its implementation, and also very powerfull. The system could also be structured so that the operator could select a particular output option as a function of the result of the evaluation.

CONCLUSION

There are currently several applications where it is desirable or even necessary to evaluate magnetic tapes of ECG's at many times real-time. In particular, PVC's are of great clinical and research interest*. Much work has been done on computer analysis of ECG's in general, but this application has not been well-investigated.

An effective solution must have high accuracy, be rapid, be flexible, and be easy to use. The general blocks that are required in such a program are input, data reduction, detection, measurement, decision, and output.

Such a system has been implemented and works, on a Nova computer. On a clinical data base including over 75 PVC's, every PVC was flagged. The number of normal beats and artifacts identified as PVC's was under 1%. The tested speed is over 8 times real-time and has been calculated to be about 14 times on the Nova. Implementation of the same program on the Nova 800 will increase the speed by a factor of 4 or 5. Flexibility and expandability are inherent in the design.

At this point, a thorough analysis of the effectiveness of the system must be done, on a larger data base. In addition to determining the actual power of the system, this will provide the basis on which to establish better values for the program parameters. It is expected that this step will greatly reduce the rate of false positives.

One option that could increase the speed of analysis by 50% is to reduce the sampling rate from 200 sps to 120 sps. The output block may be changed or added to in order to mold the system for a particular application. It is hoped that other changes will also be made without affecting the overall flow of the program, to elevate this from a working system to a useful tool.

REFERENCES

1. Black, Herbert, "Medical Pulse/Machines Monitor Heart Constantly", Boston Globe, c.April 1, 1973.
2. Bonner, R.E., "A Computer System for ECG Monitoring", IBM Technical Report no. 17-241, IBM Corp., Advanced Systems Development Division, Yorktown Heights, N.Y.
3. Caceres, Cesar A. and Dreifus, Leonard S., Clinical Electrocardiography and Computers, Academic Press, Inc., 1970, New York, N.Y. 471 pages.
4. Cox, J.R.Jr., et. al., "Some Data Transformations Useful in Electrocardiography", Computers in Biomedical Research, ed. Stacy and Waxman, vol.3, Academic Press, New York. pp. 181-206. Quote was from p.194.
5. Cox, J.R.Jr. and Nolle, F.M., "Arrhythmia Monitoring Algorithm for Real Time Application", Proceedings of the Hawaii International Conference on System Sciences-Computers in Biomedicine, pp.120-122, Honolulu, Hawaii, 1972.
6. Cox, J.R.Jr., et.al., "Digital Analysis of the EEG, the Blood Pressure Wave, and the ECG", Proceedings of the IEEE, v.60 no.10, October, 1972, pp.1137-1164.
7. Cox, J.R.Jr., et.al., "AZTEC, A Preprocessing Program for Real-Time ECG Rhythm Analysis", IEEE Transactions on Bio-Medical Engineering, v.15 no.2, April 1968, pp.128-9.
8. Feldman, C.L., et.al., "Computer Detection of Ventricular Ectopic Beats", Computers in Biomedical Research, v.3, p.666, 1970.
9. Kohn, M.S., "Characterization of Ectopic Heartbeats", SM Thesis, Electrical Engineering, MIT, 1968.
10. Kurnik, P.B., "Computer Aided Analysis of Electrocardiograms", Tech Engineering News, v.54 no.7, December 1972, pp.7-16.
11. Lown, Bernard, "Intensive Heart Care", Scientific American, v.219 no.1, July 1968, pp.19-27.
12. Lynch, J.T. "Arrhythmia Detection and Classification Using a Digital Computer on Electrocardiographic Data" SM Thesis, Electrical Engineering, MIT, June 1965.

13. Naylor, W.S., et.al., "A Reliable ECG R-Wave Detector Applicable to Normal and Abnormal Rhythms", Digest of the Seventh International Conference on Medical and Biological Engineering, p.306, Stockholm, Sweden, 1967.
14. Nolle, F.M., Cox, J.R.Jr., "How to Keep Up With the ECG", Digest of the 1969 IEEE Computer Group Conference, pp.182-185.
15. Nolle, F.M., "Arrhythmia Monitor for Cardiac Patients", 24th Annual Conference on Engineering in Medicine and Biology, Las Vegas, Nevada, p.150, 1971.
16. Pryor, T.A., et.al., "Electrocardiographic Interpretation by Computer", Computers and Biomedical Research, v.2, pp.537-548, 1969.
17. Schwann, H.P., editor, Biological Engineering, Gesselowitz, D.B. and Schmitt, O.H., "Electrocardiography", pp.333-391, McGraw-Hill, 1969.
18. Walters, J.B.Jr., "Portable Clinical or Laboratory Computing System", Quarterly Progress Report, No.106, July 15, 1972, MIT Research Laboratory of Electronics, pp.172-184.
19. Walters, J.B.Jr., "Signal Preprocessing as an Aid to On-Line EKG Analysis", SB Thesis, Electrical Engineering, MIT, 1971.
20. Wartak, Josef, Computers in Electrocardiography, Charles C. Thomas, Springfield, Illinois, 1970. Quote was from p.vii.